

# Quantum Snakes on a Plane: Mobile, Low-Dimensional Logical Qubits on a 2D Surface


Adam Siegel<sup>1,2,\*</sup>, Zhenyu Cai<sup>1,2</sup>, Hamza Jnane<sup>1,2</sup>, Balint Koczor<sup>1,3</sup>, Shaun Pexton<sup>1</sup>,  
Armands Strikis<sup>1,4</sup> and Simon Benjamin<sup>1,2</sup>

<sup>1</sup>*Quantum Motion, 9 Sterling Way, London N7 9HJ, United Kingdom*

<sup>2</sup>*Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom*

<sup>3</sup>*Mathematical Institute, University of Oxford, Woodstock Road, Oxford OX2 6GG, United Kingdom*

<sup>4</sup>*Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom*

 (Received 15 February 2025; revised 2 July 2025; accepted 17 December 2025; published 25 February 2026)

Recent demonstrations indicate that silicon-spin quantum processing unit will be able to shuttle physical qubits rapidly and with high fidelity—a desirable feature for maximizing logical connectivity, supporting new codes, and routing around damage. However it may seem that shuttling at the logical level is unwise: static defects in the device may “scratch” a logical qubit as it passes, causing correlated errors to which the code is highly vulnerable. Here we explore an architecture where logical qubits are 1D strings (“snakes”) which can be moved freely over a planar latticework. Possible scratch events are inferred via monitor qubits and the complementary gap; if deemed a risk, remarkably the shuttle process can be undone in a way that negates any corruption. This leads to high levels of tolerance against shuttling-related imperfections and enables logical operations between snakes by a semitransversal method. We conclude that this approach is suitable for fault-tolerant computing in both near-term and long-term, mature-era silicon devices.

DOI: [10.1103/494s-jd8h](https://doi.org/10.1103/494s-jd8h)

## I. INTRODUCTION

Quantum computing promises to outperform classical machines provided sufficiently low logical rates are achieved so that deep quantum algorithms can be run. Such low error rates can be obtained by assembling a significant number of qubits together, in so-called error correcting codes [1,2]. The choice of the code is often determined by the constraints that the platform imposes, for example, the surface code is well suited for devices that have 2D qubit layouts with nearest-neighbor connectivity [3,4]. Yet, even apparently simple 2D layouts may prove challenging to engineer as systems scale: a dense grid of qubits can quickly become unmanageable, due to the supporting infrastructure that is necessary for each qubit’s control and measurement. Recent studies have advocated for alternative and more experimental-friendly architectures that are enabled by qubit shuttling. Indeed when a platform supports the shuttling of physical qubits,

there are potential advantages beyond the increased flexibility of design. Examples can include support for higher connectivity low-density parity check codes, logical qubit connectivity, and routing around damage.

While several qubit platforms have been identified as suitable for implementing shuttling with high fidelity [5,6], this paper will mostly focus on silicon-spin qubits. Indeed, on top of demonstrated reliable shuttling [7–9], these qubits are also an excellent candidate for the implementation of a quantum computer. Fast and high-fidelity gates [10–14] and high coherence times [15] have been shown. Besides, their minimal footprint ( $100 \times 100 \text{ nm}^2$  per qubit) guarantees high scalability, yielding moderately sized quantum computers, even in the million-physical-qubit regime. Interestingly, it is also expected that a dynamic quantum computer continuously shuttling its electrons would prove beneficial, showing reduced dephasing noise [16], motional averaging, and a synchronization of qubit frequencies for gates implementation [17]. Appendix A describes the specificities of silicon-spin qubits in more detail.

Recent works proposed and explored an extremely simplified shuttling-based silicon-spin-qubit architecture: a  $2 \times N$  array of qubits whose first row can be collectively shuttled with respect to the second row [18,19]. Although the linear architecture imposed various constraints, it was

\*Contact author: [adam@quantummotion.tech](mailto:adam@quantummotion.tech)

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

shown that powerful code classes could still efficiently be embedded in such a paradigm. Full universal quantum computation was shown to be possible using a surface code encoding, making this approach a feasible candidate for early fault tolerance.

This architecture however suffered two main drawbacks. First, the highly restricted dimensionality induced connectivity lockups at the logical level, meaning that in general two-qubit logical gates could not be implemented in parallel, thus drastically increasing the runtime. For instance, a simple  $6 \times 6$  Hubbard model would take 2.5 days to run [18]. Second, the proposed architecture was not defect-resistant. If a qubit site were to become inoperable sometime during the computation, preventing qubits from shuttling through it, the device would simply be split into two disconnected halves, rendering it unusable.

In this paper, we take a step forward and introduce an architecture suited for large-scale fault tolerance. It is based on a 2D *latticework* formed of low-dimensional array strands that occasionally meet at junctions or run parallel. The model is tailored to silicon-spin qubits, although other platforms which support shuttling may also

be compatible with the concepts. There are a wide range of possible latticework structures; the strands can be very long, creating large void areas to accommodate support systems including measurement devices.

Latticework structures of this general kind have been considered in Refs. [18–25], but the paradigm for fault-tolerant computation was relatively conventional. Indeed, in all these prior works, a logical qubit is laid out as a grid corresponding to the canonical surface code, with shuttling (or SWAP gates in the case of Ref. [25]) used to bring data and ancilla qubits together. See Figs. 1(d) and 1(e).

In the present “Snakes on a Plane” (SoaP) model, we instead take a more radical approach, where the logical qubits are snakelike objects, or trains of physical qubits that move through the latticework. Shuttling, combined with junctions for rerouting the snakes, allows for the global movement of the logical qubits at the device scale and the exclusion of potentially defective links (see Fig. 1, left panel). This ability to globally displace logical qubits is a fundamental novelty of the current proposal, and was incompatible with previous approaches (Fig. 1, right panels). Ideas from the  $2 \times N$  paradigm [18] are only

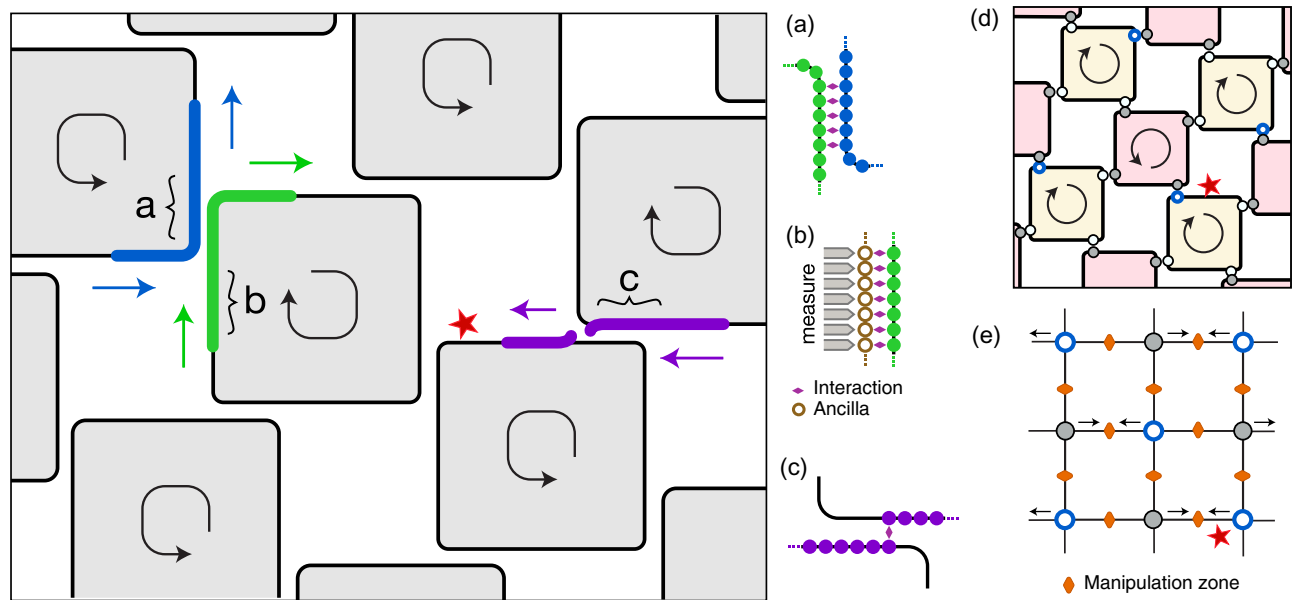


FIG. 1. Paradigms for fault-tolerant quantum computation in a 2D latticework, exploiting shuttling of physical qubits along linear arrays (black lines) which in places run parallel and permit interactions. The “Snakes on a Plane” approach is shown in the main panel and subfigures (a)–(c). Here a logical qubit is a 1D chain of data qubits, local on the scale of the latticework structure. Stabilizers are evaluated by small to-and-fro shuttling motions with respect to fixed ancilla and measurement elements (part b and see Fig. 3). In the figure, the purple snake is moving from one loop to another, e.g., via SWAP operations in the parallel-array region marked (c). Meanwhile the blue and green snakes are performing a semitransversal logical gate in the region marked (a). In contrast, prior schemes leveraging shuttling such as the “looped pipeline” approach [shown in panel (d), cf. Fig. 10 of Ref. [20]], typically lay out data qubits in a grid matching the canonical surface code and do not involve net motion of the logical qubit. Other previous approaches like the SpinBus architecture [21] or the spiderweb array [22] [shown in panel (e)] follow the same simple approach and only utilize shuttling to bring data and ancilla qubits to manipulation zones where they can be initialized, measured or operated on. While the Snakes on a Plane paradigm has advantages in connectivity, a potential risk is the impact of a phase-shifting defect in the device (red asterisk). Without a suitable protective protocol, this may “scratch” an entire logical qubit as it moves past and thus induce a logical error. The purple snake will imminently encounter this risk.

borrowed for the stabilization of the snakes. Assuming shuttling fidelities that are only slightly better than currently demonstrated ones, we there showed (see Fig. 9 of Ref. [18]) that these linearized logical qubits are practical memory qubits as they can reach error rates as low as  $10^{-14}$ . Now, although the 2D latticework is only an incrementally more challenging technological scenario, the resulting approach is profoundly superior to the  $2 \times N$  array in terms of connectivity and damage tolerance. However, in comparison to prior models there is now a significant new problem to address: by shuttling logical qubits through a solid-state environment there is the risk that static defects in the device will “scratch” the logical qubit, i.e., create correlated strings of errors against which the code is vulnerable (red stars in Fig. 1). We introduce an entirely novel protocol to mitigate this risk.

In Sec. II, we present in more depth the general ideas and layouts underlying the *Snakes on a Plane* proposal. The following Sec. III is the largest part of the paper: here we review the noise processes that dominate in silicon-spin qubits and describe a means to mitigate their impact on shuttled logical qubits. We give particular attention to charge noise, which originates from the random fluctuation of charges between defects spread across the device and effectively induces uncontrolled variation in the magnetic field landscape. These can be responsible for catastrophic dephasing errors affecting entire logical qubits shuttled near the defect, but we describe a protocol that is highly effective in mitigating the impact of such events. The approach is enabled by two defect-detection schemes: one is based on the continual evaluation of the effective magnetic field landscape via the use of *monitor qubits*; the other analyses the stabilizer measurements signature, more precisely by computing the syndrome’s complementary gap. The capability to infer likely defect-related events enables a robust shuttling protocol which we call *snake surgery*, allowing one effectively to reverse time and retrieve the logical qubit’s state before the defect occurred. The joint use of these methods enables the shuttling of logical snakes across long distances without accumulating errors. In Sec. IV, we show that universal quantum computation is possible and efficient on our proposed architecture. In particular, the long-distance-shuttling ability enables an all-to-all connectivity at the logical level, where two-qubit gates are efficiently implementable transversally, or *semitransversally*. The latter protocol, which we describe in Sec. IV C, promises to reduce logical error rates if accepting higher time costs than for usual transversal gates, in case the noise level is prohibitively high.

This paper therefore represents a considerable leap forward in the implementation of large-scale silicon-based quantum computers. By capitalizing on demonstrated high-fidelity shuttling, we design an entirely novel architecture that can achieve logical error rates that are practical for running deep quantum algorithms. It features many

empty spaces, which are highly desirable for fitting control electronics. The dynamic picture we are considering, where the logical objects are assimilated to 1D snakes, enables a highly desirable all-to-all logical-level connectivity, which was not supported by any existing proposal, and the capability to route around damage. Gates are implemented transversally rather than by lattice surgery, which further reduces the runtime. The only requirements of this proposal are fast and high-fidelity shuttling (which was already required by previous latticework schemes) and a moderate resource overhead, at most a factor 2, which is a small price to pay given the minimal footprint of silicon spins. This is to accommodate for monitor qubits, a necessary ingredient in guaranteeing the integrity of the snakes when shuttled in a solid-state environment.

## II. GENERAL IDEA

In this section, we present the general layout of the snakes on a plane architecture and highlight in general terms where its advantage arises. For a more detailed breakdown behind our choice of qubits, see Appendix A.

### A. Error correction with a $2 \times N$ array of qubits

We here review our previously published work [18], which is incorporated into the current paradigm. In this paper, we analyzed the suitability of  $2 \times N$  arrays of qubits equipped with shuttling for embedding error correcting codes. Such  $2 \times N$  arrays, simply consisting of two parallel rows (or “rails”) of physical qubits, are the building block of our current architecture and are represented in Fig. 2. In this figure, the first row of qubits can be collectively shuttled while the second row is kept static. Two-qubit gates can be implemented between neighboring qubits across the rails. By placing data qubits in the mobile rail and ancilla qubits in the static one, all-to-connectivity is virtually obtained between data and ancilla qubits thanks to shuttling: this theoretically permits the implementation of any code. Nonetheless, as shuttling is a finite-time and

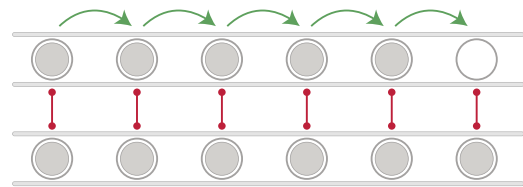


FIG. 2. Representation of the  $2 \times N$  architecture. The device is characterized by two parallel rails of evenly spaced qubits (gray circles with gray disks inside). Adjacent qubits from different rails are allowed to interact via two-qubit gate operations (red vertical lines with disks on their ends). Finally, the qubits of the first row are allowed to shuttle along their rail (green arrows). Some empty locations are kept at the end of the device to facilitate movement (empty gray circle).

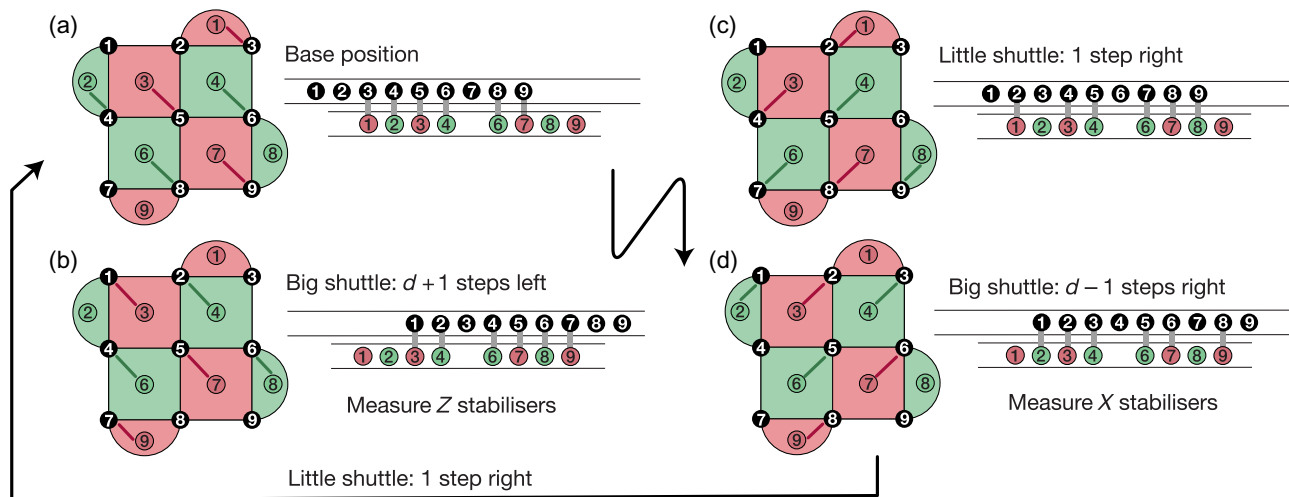


FIG. 3. Implementation of a stabilizer cycle on the Snakes on a Plane architecture. In each panel, we represent the canonical 2D layout on the left, and a schematic for the layout on a  $2 \times N$  array on the right. Here, red and green colors indicate  $X$  and  $Z$  stabilizers respectively. Interactions are enabled by shuttling the data qubits' row back and forth and interacting the qubits with corresponding nearest-neighbor ancillas. The stabilizer can be implemented in a total shuttling length of  $2d + 2$ , where  $d$  is the code distance. Note that the ordering of the shuttles and the staggered measurement of the  $X$  and  $Z$  stabilizers protects from hook errors, as shown in Ref. [18].

noisy process, one should aim at minimizing the overall shuttling distance over a stabilizer cycle. In Ref. [18], we demonstrated that an appropriate choice of qubit labeling along with the right gate and shuttling sequencing leads to advantageous performances for certain code classes. The rotated surface code is one of them and is the code we will focus on for the rest of this paper. A schematic representation of a  $3 \times 3$  surface code embedded on a  $2 \times N$  array is given in Fig. 3.

As noted in Ref. [18], the use of a 1D embodiment for the canonically 2D surface code logical qubits has the effect that asymptotically the code cannot scale: for any given finite shuttling speed there is some logical qubit so large that a stabilizer cycles cannot be completed before the accumulated noise is too great. However, the asymptotic performance may not be relevant to the practical utility: attainable speeds and fidelities for shuttling mean that this problem need not arise for logical qubit sizes of practical interest, even those with distance 30 or more. This is explored further in later sections.

An important characteristic of the rotated surface code embedded in the  $2 \times N$  architecture is that stabilizer cycles can be implemented in only four shuttles of total length  $2d + 2$  (where  $d$  is the code distance). This is visible in Fig. 3: South-East and North-West data qubits around an ancilla are separated by  $d + 1$  increments (top right and bottom right panels). Shuttling back and forth along these  $d + 1$  increments allows for all interactions within the stabilizer cycle and for the data qubits to return to their initial position. We also showed in Ref. [18] that this implementation protects against hook errors. This term refers to a process that can potentially occur during a stabilizer cycle,

whereby a single gate failure leads to two data qubit errors. It is a potential issue in any device using the surface code or indeed any code involving stabilizer cycles composed of multiple gates. The relevant scenario is that a gate operation between a data qubit and the ancilla qubit corrupts both, and the ancilla qubit error subsequently propagates to another data qubit in a later step of the cycle. If this leads to, for example, a pair of  $X$  errors lying along the direction of the  $X$  logical qubit, then it is “twice as bad” as a single error in terms of likelihood of contribution to a logical error. This issue can be mitigated by properly ordering the stabilizer cycle so that when a gate failure leads to two data qubit errors, they must lie in the orthogonal direction. Alternatively one can extend the size of the logical qubit according to the risk of such error pairs (which may be small).

## B. Snakes on a Plane architecture

In Ref. [18] we gave an end-to-end protocol for universal quantum computation using one long  $2 \times N$  array where all logical qubits were appended next to each other. Long-range interactions between logical qubits were then implemented via the use of a logical ancilla bus rather than by physically bringing the logical qubits closer together. While promising for the early fault-tolerant regime, this proposal suffered two drawbacks presented in the introduction: a limited connectivity between logical qubits and a high sensitivity to single-qubit failure.

In this paper, we consider a new paradigm, where logical qubits can be displaced across a latticework structure of  $2 \times N$  filaments (Fig. 4) meeting at occasional three-way

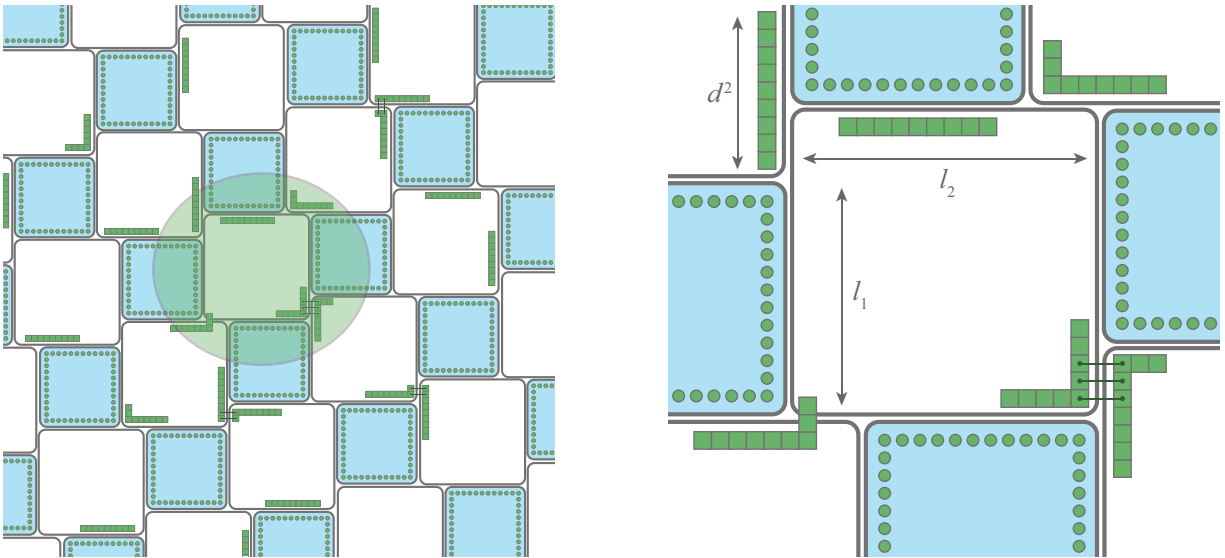


FIG. 4. An example layout for the snakes on a plane architecture. Each edge is a  $2 \times N$  filament as in Fig. 2. Data qubits (green squares) can be shuttled along the edges of a white square, while ancillas (green circles) are static and placed along the edges of a blue square. Logical qubits are built from multiple data qubits, creating a snake that can be shuttled across the device. The left panel is a zoomed-out schematics of the device. The right panel is a close-up of the region inside the green faded oval. The two bottom right logical qubits are interacting through two-qubit transversal gates, while the four other logical qubits are normally being stabilized and shuttled.

junctions. These filaments form loops of two kinds: small loops (of size  $l_1$ ) holding ancilla qubits and larger loops (of size  $l_2$ ) holding data qubits. The sizes are expressed in the number of quantum dots. Additionally, one can distinguish two kinds of edges on the lattice: data/ancilla and data/data edges. These will respectively be denoted as *stabilizer edges* and *interaction edges*. Indeed, in the former, data qubits are facing ancilla qubits, which can be utilized to measure stabilizers. In contrast, in the latter, while stabilizers cannot be measured, physical qubits from two distinct logical qubits can meet and interact across the ridge. In our architecture, data qubits are allowed to shuttle (while ancilla qubits stay fixed). Therefore, for the interaction edges, shuttling is permitted on both rails. The size of the interaction edges is  $l_{\text{int}} = l_2 - l_1$ .

In this representation, each logical qubit is thus embodied as a snake that can travel long distances. Denoting the code distance as  $d$ , each snake has length  $d^2$ . When no logical operation needs to be implemented, a snake can be shuttled back and forth along a stabilizer edge in order to measure stabilizers in the manner of Ref. [18] [Fig. 5(a)]. When the snake must be moved to a specific location of the chip, e.g., to interact with another snake, it can simply be moved forward [Fig. 5(b)]. Note that doing so may however introduce hook errors. Therefore, it may be preferable to implement such forward motion separately from stabilizer cycles. For instance, if shuttling is low-noise and fast compared to ancilla qubit measurement times, the information can be stabilized via the sequence of Fig. 5(a), and the data qubits moved forward while ancillas are being

measured. When a snake eventually reaches a junction, the user can decide on the path it will follow, either keeping it the current loop or transferring it to the neighboring one (Fig. 6). Note that the implementation of junctions that snakes can jump across can be circumvented, in case it represents a significant experimental challenge. An alternative solution yielding the same net result is to bring a *blank* snake initialized in, say, the  $|0\rangle$  state, to the other side of a junction and swap it with the current snake (similarly to the bottom right scenario of Fig. 4, right panel).

In general terms, one can easily understand where the strength of this architecture comes from. Assuming fast and high-fidelity shuttling, the snakelike logical qubits can reliably be moved across the device, enabling all-to-all

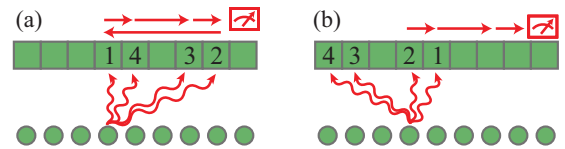


FIG. 5. Stabilizer measurements while shuttling a snake. Green squares (resp. circles) represent data (resp. ancilla) qubits. Wiggly arrows connect a given ancilla with the qubits it must entangle with for a subsequent stabilizer measurement; this occurs when the qubits become aligned. Straight arrows show the snake movement necessary to implement a stabilizer cycle, which ends with the measurement of an ancilla qubit. (a) Stabilizing a snake with no net shuttling. (b) Stabilizing a snake with a global forward movement.

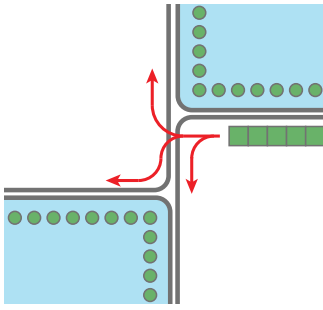


FIG. 6. Representation of a junction. Red arrows show the three potential directions a snake can take.

connectivity at the logical level. Additionally, transversal gates rather than lattice surgery may be carried out through the interaction edges, thus accelerating the implementation of two-qubit gates. Finally, the device is resilient against defective edges, as long as they can be detected (see Sec. III): if a chosen path has been deemed as faulty, snakes can simply be rerouted and travel through a different path (Fig. 7).

Note that other latticework structures are conceivable for our proposal. A hexagonal lattice and a rectangular lattice are for instance sketched in Appendix C.

### III. PROTECTION AGAINST CHARGE NOISE

As the impact of nuclear spins can be reduced by isotopically purifying devices [26], the most detrimental source of noise for silicon-spin qubits is charge noise [14,27–33]. If not dealt with properly, it can lead to out-of-calibration drifts or catastrophic events that would hinder the performance of our error correcting protocols. In this section, we first give a description of the noise processes we envision, before providing four distinct protocols that can be used jointly so as to reach exponentially low logical error rates.

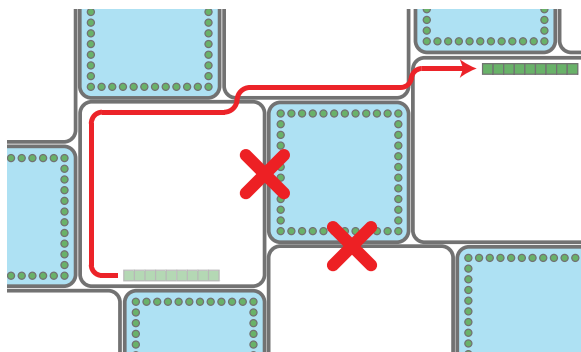


FIG. 7. Rerouting of a snake in the presence of defects. The initial and target positions are indicated by a faded and brighter snake respectively. The red arrow represents a possible path between these positions and that avoids the defective shuttling links (red crosses).

The first one relies on the choice of a qubit encoding that is resilient against phase errors. The second method is a fault-tolerant protocol allowing one to, in effect, *reverse time* and go back to the state of the system before a catastrophic phase event occurred, if such an event is suspected. Detection is enabled by the last two schemes, which respectively exploit additional monitor qubits and the analysis of the syndrome signature to infer the presence of a defect.

#### A. Noise description

While the physical origin of charge noise remains a matter of debate and is device-dependent, models based on two-level fluctuators have shown good agreement with experiments [28]. These fluctuations are typically characterized by a  $1/f$  noise spectrum which will lead to a slow drift of the  $g$ -factor landscape. Whether the electrons are shuttled or not, this drift needs to be characterized, or the electrons would progressively acquire some relative phase with respect to each other, leading to dephasing noise [27,31,34]. A problematic aspect of charge noise is its dynamic nature: a defect can arise during computation and persist for a short or long period, or even be permanent.

A highly damaging situation could occur if a defect appears near the shuttling tracks and its local impact cannot be entirely encapsulated in the above description. Rather, we will suppose that it will lead to some phaselike error being applied to all qubits flowing near the defect. To encompass the most problematic cases, we will suppose that the phase shift imparted by such effects is arbitrary, potentially way above the surface code threshold, and can occur abruptly. One can thus easily see the dramatic impact these defects might have when shuttling an entire logical qubit next to them, as they can effectively *scratch* all the physical qubits of the code. Indeed, if the imparted phase shift were close to  $\pi$ , the scratch event would directly give rise to a logical phase flip for odd-distance surface codes. This process is sudden and could lead to orbital excitations, and thus to dephasing due to the difference in  $g$ -factor between the orbital states. Fortunately, given that the difference in  $g$ -factor between the states is small and the fast decay rate of the excited states, we expect that this effect will not be the most common source of dephasing. Nevertheless even rare events of this kind could create a floor in the achievable logical error rates; a scalable fault-tolerant architecture must actively remediate the issue.

The general description we will adopt is thus the following: each location in the device will be associated with a phase channel, parametrized by the local  $g$ -factor. Charge noise, as described above, can lead to fluctuations of these channels of two types. The first model gives rise to slow and small variations of the magnetic field felt by the electrons, while the second more local description can lead to sudden and severe phase errors. Our aim is to mitigate the impact of both these contributions. In the remainder of this

section we characterize the severity of the more ubiquitous slow and small phase variations.

By definition, the rate of appearance of the fluctuators yielding the phase variations is at least of the order of  $T_2^*$ , whose demonstrated values reach up to  $20 \mu\text{s}$  [14,26]. This is an order of magnitude higher than stabilizer cycles times (a few microseconds, see Appendix A): we thus expect few variations at this time scale. Of course, variations that are slower than the whole computation time will be accommodated by calibration beforehand and application of unitary corrections during the computation. Alternatively, one could use a dynamical-decoupling-based scheme allowing a snake to pass through the same phase channel multiple times under a different basis to cancel out the noise. Therefore, the only timescale that is relevant to study is that of phase fluctuations that may arise in the middle of the computation or evolve before its end.

### B. Encoding the qubits in a decoherence-free subspace

When a single qubit is represented by a single spin, one obtains the so-called Loss-DiVincenzo (LD) encoding. Although this is the original and the most natural encoding [35], quantum information can also be stored over multiple spins [16]. This provides interesting features, such as additional protection through the use of a decoherence-free subspace when using singlet-triplet (ST) encoding [36], and full electrical control when using exchange-only (EO) qubits [37], which are two- and three-spin encodings respectively.

As explained before, fluctuations of the  $g$ -factor landscape due to charge noise represent an important source of errors for silicon-based quantum computers. For conventional quantum processor architectures, these fluctuations affect individual physical qubits. However, as long as these effects remain below a certain threshold, error correction will suppress their impact on the computation. Since different encodings behave differently in fluctuating environments, our goal is to determine which encoding is better suited for our architecture. While noting that a carefully chosen encoding can mitigate the effect of a sudden apparition of a charge near the shuttling track, we focus here on mitigating the effect of the slowly varying  $g$ -factor landscape. The techniques presented in the next subsections will be targeted toward reducing the impact of catastrophic *scratching* events. Here, we focus on the comparison between LD and ST qubits as performing gates on EO qubits remains challenging [16].

Recently, the authors of Ref. [38] explored this comparison under the assumption that the environmental changes induce a randomly fluctuating magnetic field in space and time. By performing a similar study as in Ref. [38], one can extract the dephasing factor  $\mathcal{W}$ —which can be understood as a measure of the shuttling fidelity—due to the fluctuations of the magnetic field as a function of the shuttling

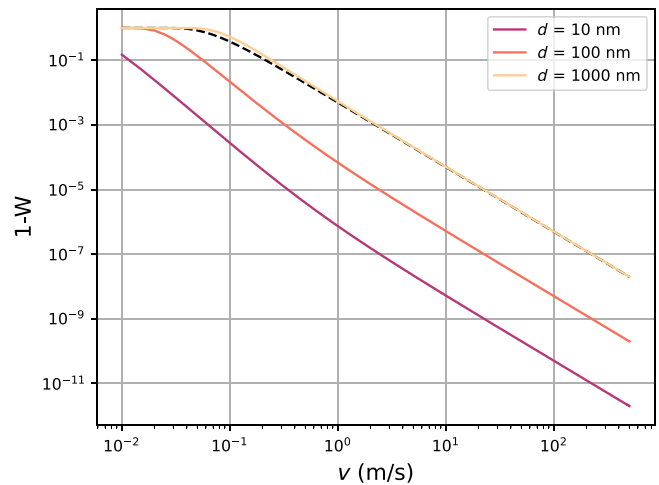


FIG. 8. Dephasing error as a function of the shuttling speed for different qubit encodings. The qubit is shuttled for a distance  $L = 10 \mu\text{m}$  and a time  $T = L/v$ . As the distance  $d$  between the electrons composing the singlet-triplet qubit increases (solid lines), the probability for them to experience a different field at a given point increases, leading to a higher infidelity. Within this noise model, the singlet-triplet always outperforms the Loss-DiVincenzo for reasonable values of  $d$ . One would require a distance  $d > 1 \mu\text{m}$  for the Loss-DiVincenzo (black dashed line) encoding to be better. These curves are obtained using the analytical expressions derived in Ref. [38].

time  $T$  for both encodings. Considering a shuttling length  $L = 10 \mu\text{m}$  and a time  $T = L/v$  with  $v$  the shuttling speed, we plot the evolution of the infidelity  $1 - \mathcal{W}(T)$  for LD and ST qubits. More details on this model are given in Appendix D.

As the speed increases in Fig. 8, a qubit spends less time shuttling and therefore experiences less magnetic field fluctuation. This explains why, irrespective of the encoding, the infidelity decreases monotonically with speed. The fact that the ST encoding (solid orange and purple lines) outperforms the LD encoding (black dashed lines) for practical values of  $d$  can be explained as follows. When using a ST qubit, the state is protected against situations in which both electrons experience the same change in magnetic field. In Fig. 8, we benefit from this protection and the remaining infidelity comes from the fact that the two electrons are separated by a small distance, and thus feel slightly different  $B(x, t)$ . As  $d$  is reduced, the difference in the magnetic field experienced by both electrons decreases and so does the infidelity. However, as  $d$  gets larger, there comes a point where both electrons are too far apart to experience a similar field, leading to a larger infidelity. For distances  $d = 1 \mu\text{m}$  (solid yellow line), we even find that the LD encoding performs better. In Appendix E, we report the evolution of  $1 - \mathcal{W}$  with speed for different values of the correlation length  $\lambda$  and correlation time  $\tau$ , which confirms our understanding.

Within this noise model, the infidelity keeps decreasing with speed. However, it does not account for charge noise, spin-orbit coupling, or valley effects which are known to be detrimental to shuttling at high speed [39]. Therefore one needs to be careful before making definitive conclusions. Nevertheless, since spin-orbit-induced effects and valley state excitations are believed to be negligible at slow speeds, this study suggests that ST qubits are better suited in this regime. For larger speeds, a more realistic noise model would be required to conclude.

The remaining dephasing error when using the ST encoding stems from the fact that the two electrons cannot exactly experience the same field as one will always be in front of the other. While further suppression may be possible with additional steps, for the present paper we simply note that the achievable dephasing error is very low at the likely shuttling speeds of 10–50 m/s. Consequently this aspect of shuttling noise, while ubiquitous in the device, will not be a dominant noise source in well-designed stabilizer cycles involving modest shuttling distances (as employed here). In essence, the decoder handles these errors along with sources such as gate infidelity. We therefore now turn to consider the more rare, but more problematic case of defects that can “scratch” logical qubits.

### C. Snake surgery

As described in Sec. III A, we assume here that the appearance of charge defects gives rise to unwanted phase effects. If a charge defect materializes near a shuttling track, a sudden and severe phase error can affect the data qubits that are shuttled through it, alarmingly increasing the logical error probability. While the ST qubit encoding presented in the previous section should reduce the impact of the resulting error surge, additional protection may be necessary to bring it to the extremely low levels required to run deep quantum algorithms. In this section, we suppose that catastrophic events are phaselike and can be detected (we give in Secs. III D and III E protocols that are able to do so). Importantly, while detection needs to be highly *sensitive*, i.e., able to flag genuine error events reliably, it need not be highly *specific*: it is not problematic if the majority of flagged cases are in fact error-free. With these assumptions, we present a fault-tolerant scheme allowing one to retrieve the logical qubit state prior to any corruption that may have occurred, removing its impact regardless of how strong its effect was.

Let us assume that a localized “pin” charge did appear near a shuttling track. Two different cases need to be distinguished, depending on whether a given snake is merely being stabilized in place with modest back-and-forth motion or shuttled across a long distance [Figs. 5(a) and 5(b)]. The first situation can easily be dealt with by noticing that during the stabilizer cycles, a snake is only

being shuttled back and forth by  $d + 1$  increments [as in Figs. 3 and 5(a)]. Assuming that the shuttling direction is, say, along the columns of a surface code, a pin defect would only *scratch* one single column of the code. By choosing to orient the  $Z$  logical operator perpendicularly (along the rows), the code would naturally be protected against such adversarial phase errors. Note however that, while exponential error suppression is still guaranteed, the logical error rate and threshold may be moderately affected by the presence of defect-induced columns of  $Z$  errors, albeit perpendicular to the  $Z$  logical operator. Indeed, these strings offer more opportunities for actual  $Z$  logical errors to form, both by slightly reducing the code distance and increasing the number of error paths.

Consequently, the scenario one should carefully scrutinize is that of Fig. 5(b), where a snake is shuttled far away from its original position. This kind of movement could indeed lead to errors affecting all data qubits, potentially beyond the error threshold of the surface code. More specifically, we want to protect a snake from a defective shuttling link that the monitoring techniques (see Secs. III D and III E) detected only *after* the snake shuttled through it. If a defect is detected before the snake reaches the intended shuttling path, it can be simply rerouted.

This kind of situation can be tackled by the protocol depicted in Fig. 9, which we call *snake surgery*. It relies on the transfer of the logical information contained in one

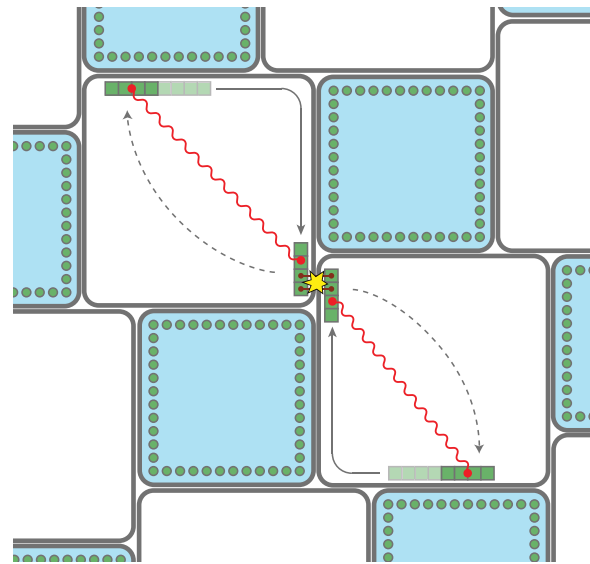


FIG. 9. Snake-surgery protocol. During long shuttles, snakes are prone to *scratch* events, i.e., catastrophic phase changes along the shuttling tracks due to the appearance of pointlike particles. Fault tolerance is guaranteed by only shuttling the head of a snake (solid arrow) while the tail is stabilized in place. Entanglement is maintained between them throughout the shuttling process (red wiggly line). If a catastrophic event was detected by the monitor qubits or stabilizer measurements (yellow star), the information can be teleported back to the tail (dashed arrow).

snake onto two distinct but entangled snakes, called *head* and *tail*. These are obtained by successively growing and cutting the original snake via lattice surgery [40]. While the tail remains static and retains full error-correction capabilities, the head is shuttled across long distances. If no defect is *a posteriori* detected, the tail is destroyed, projecting the delocalized logical state onto the head: this process thus correctly transfers the information to the remote location in the device. On the contrary, if a defect was suspected, the head is measured out, which will project the logical state back to the tail. Importantly, this will succeed regardless of whether a defect event in fact occurred. The protocol can then be restarted through another shuttling route or delayed until the defect has disappeared or become stable.

Further, this analysis not only holds when shuttling across long distances, but also when two distinct logical qubits interact. Specifically, one would follow the same protocol, i.e., stabilizing the tails of both logical qubits in place, while shuttling the heads so as to bring them to both sides of an interacting edge. They can then be interacted through transversal CNOTs (or via the semitransversal protocol of Sec. IV C). As such interaction requires shuttling across the interaction edge, defect-detection protocols must assess if this edge became defective or not throughout the process. Depending on the outcome of such monitoring, the heads or tails are measured out as per the above protocol.

A more detailed description of the scheme can be found below and in Fig. 10, in the case of shuttling one snake across long distances. The case of two interacting snakes is left to Appendix F.

### 1. Step 1

Double the length of the logical snake. This is performed by initializing additional data qubits in the  $|0\rangle$  state and performing an  $XX$  lattice surgery between the original snake and a logical snake prepared in the  $|0_L\rangle$  state. Note that because  $X_L$  is parallel to the direction of the shuttling, this  $XX$  lattice surgery can be performed efficiently, i.e., by shuttling the qubits over  $O(d)$  dots for each round of the lattice surgery [in Fig. 10(a) the last  $d$  qubits of the first snake and the first  $d$  qubits of the second snakes are both  $X_L$  logical operators]. This is the same shuttling distance as for the implementation of normal stabilizer cycles. Starting from the initial state  $|\psi\rangle$  encoded in a  $d \times d$  surface code, this step therefore encodes the same state  $|\psi\rangle$  in a  $2d \times d$  surface code (where  $2d$  is the length of the  $Z$  logical operator),

$$|\psi\rangle_{d \times d} \rightarrow |\psi\rangle_{2d \times d}. \quad (1)$$

### 2. Step 2

Split the extended surface code into two separate logical qubits, the *head* and the *tail*. The split operation is

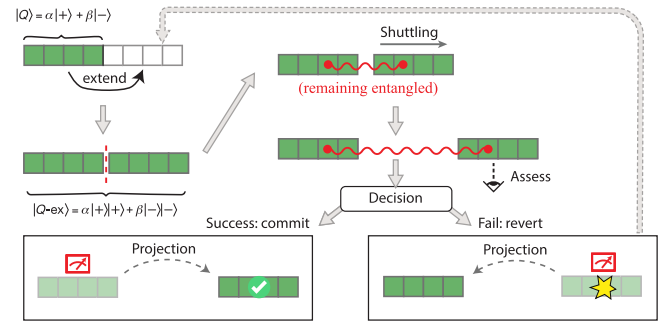


FIG. 10. Detailed steps of the snake-surgery protocol. (i) The length of a snake is doubled by initializing new data qubits in the  $|0\rangle$  state and performing an  $XX$  lattice surgery. (ii) The double-length snake is split in halves by performing another  $XX$  lattice surgery. This separates the head from the tail but generates entanglement between the two. (iii) The tail is stabilized in place, thereby protecting the logical information, while the head is shuttled across a long distance, making it sensitive to *scratch* events. Entanglement between the head and tail is maintained throughout the process. (iv) At the end of the head's shuttling, its integrity is assessed by analyzing its syndrome signature and making use of the monitor qubits (Secs. III D and III E). (v) If no catastrophic phase change was detected, the tail is measured out, projecting the logical information onto the head. Otherwise, the head is destroyed, projecting the information back to the tail. The procedure has to be restarted from step (i).

again lattice-surgery-based and along the same boundary. Assuming without loss of generality that the previously measured  $X_L X_L$  was  $+1$ , this results in

$$|\psi\rangle_{2d \times d} = \alpha|+\rangle + \beta|-\rangle \rightarrow \alpha|++\rangle + \beta|--\rangle, \quad (2)$$

where the resulting states are  $d \times d$  surface codes.

### 3. Step 3

Stabilize the tail in place while shuttling the head across a long distance. Throughout this process, the physical qubits of the latter can acquire some  $Z$  errors, which we denote as  $\mathcal{D}_\varphi[\cdot]$ ,

$$\alpha|++\rangle + \beta|--\rangle \rightarrow \alpha|+\rangle \mathcal{D}_\varphi[|+\rangle] + \beta|-\rangle \mathcal{D}_\varphi[|-\rangle]. \quad (3)$$

### 4. Step 4

At the end of the head's shuttling, detection techniques reassess the viability of the shuttling route. If no defect was detected ( $\mathcal{D}_\varphi[\cdot]$  is close to the identity), the tail is measured out in the  $Z$  basis (which amounts to measuring all its data qubits in the  $0/1$  basis). Depending on the measurement outcome  $m$ , the head will be projected onto

$$\alpha|++\rangle + \beta|--\rangle \rightarrow \alpha|+\rangle + (-1)^m \beta|-\rangle. \quad (4)$$

One can then apply an  $X$  gate when  $m = 1$ , thereby correctly transferring the initial state  $|\psi\rangle = \alpha|+\rangle + \beta|-\rangle$  to the head.

If a defect was detected however, the head must be measured in the  $Z$  basis to reverse the process and teleport the information back to the tail. As  $\mathcal{D}_\varphi[\cdot]$  commutes with such  $Z$  measurement, the calculation is identical to the previous case.

The above protocol therefore promises to fully suppress the impact of catastrophic events, provided they are pure  $Z$  errors and can be detected with high probability by the defect-detection schemes of Secs. III D and III E. If this is the case, the probability of correctly performing each step of our protocol (snake growth and split and teleportation of the information to the head or the tail) is simply given by the logical error rate of a  $d \times d$  surface code. Indeed, all these steps are fault tolerant, e.g., based on lattice surgery.

The time overhead induced by this protocol is simply that of the lattice surgery of steps 1 and 2, i.e.,  $d$  stabilizer cycles for the merge and one for the split.

Note that there exists a natural extension of the scheme we here presented. Instead of generating a single head and tail per snake, one could envision initializing a multi-headed snake (which one might call a hydra), in the state

$$|\psi\rangle = \alpha|+\rangle|+\dots+\rangle + \beta|-\rangle|-\dots-\rangle, \quad (5)$$

where the first qubit is the tail's state, and the other qubits correspond to the heads. While the (single) tail would still be stabilized in place to preserve the target error rate, heads can be shuttled separately. This has a dual use. First, in the case of a high defect rate, all heads could be brought to the same location in the device, yet through different routes, thereby multiplying the chances that at least one path was stable. Second, if the defect rate is instead relatively low, one could send the heads to different locations and exploit their entanglement [Eq. (5)] to optimize quantum circuits by quantum fan-out [41].

## D. Monitor qubits

### 1. Generalities

In the previous section, we demonstrated that snakes are resilient against phaselike defects, no matter how strong they are, as long as they can be detected. In the present section, we give a first protocol enabling such detection. More specifically, we detail the use of monitor qubits [42,43] that are employed to validate in real time the quality of shuttling links and to build an accurate model of their characteristics over the course of repeated shuttling events. These will be used jointly with stabilizer measurements, whose signature can as well inform the user about errors accumulated by shuttled data qubits (Sec. III E). Note however that the sole use of stabilizers cannot be sufficient, as

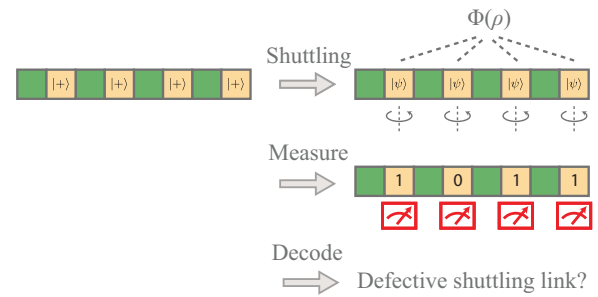


FIG. 11. Schematic representation of the monitor-qubit-based scheme for task (i), i.e., inferring the presence of a defect. Monitor qubits (light orange squares) are interleaved in between data qubits (green squares) so as to probe the phase environment the logical snake evolves in. Step 1: monitor qubits are initialized in the  $|+\rangle$  state. Step 2: the logical snake is shuttled. Every qubit picks up a phase along the way, modeled by the channel  $\Phi$ . Step 3: monitor qubits are measured in the  $X$  basis. Step 4: the measurement outcomes are used to infer if a defect occurred. The workflow is similar for task (ii), i.e., precisely estimating the phase channel  $\Phi$ .

a  $\pi$  rotation on all data qubits of a snake (induced by a pin defect) would yield a logical error with no stabilizer signature for odd-distance surface codes.

For this purpose we assume that a set of  $N$  monitor qubits are interlaced with the data qubits and that before a shuttling event, they are prepared in an initial quantum state  $\rho$ . The qubits are then shuttled which ideally should act as an identity channel on the monitor qubits but due to imperfections, this channel will be nontrivial which we model via a completely positive and trace preserving map  $\Phi$ . Our aim is to learn characteristics of this channel in real time by applying a decoding operation on the state  $\Phi(\rho)$  and measuring in the standard basis. The measurement basis obtained via the decoding operation, and the estimator used, crucially affect the performance of the approach. The overall procedure is represented in Fig. 11.

As motivated by Sec. III A, we will first assume that the noise  $\Phi$  acts as a pure unitary single-qubit  $Z$  rotation of angle  $\omega$  on all shuttled qubits. Monitor qubits must respond to two tasks: (i) detect with high precision when  $\omega$  is larger than a given value  $\omega_{\max}$  (this is a decision problem); and (ii) estimate the value of  $\omega$  with low uncertainty. The first task is the most crucial as it can inform the user of the presence of catastrophic events due to pin defects, which can induce logical errors. The second task is a refinement of the previous one, yet is less vital. It can be utilized as a means to update our knowledge of a given shuttling link for subsequent calibration. As an example of these two tasks, let us suppose that a defect appears at a given time and that a snake with interlaced monitor qubits is shuttled in its vicinity. This defect must be detected with high probability by task (i), informing the user that the shuttling event should be reversed (using the protocol of Sec. III C). The value

of the phase shift induced by the defect does not matter at this stage, as one only needs to know whether a defect occurred. If further monitoring confirms the presence of a defect, one can temporarily deactivate the corresponding shuttling link, thereby forbidding any snake from using this route. Eventually, if later monitoring shows that the defect has become stable, inducing a constant known phase shift [task (ii)], then qubits can again be routed through this link.

We detail in Appendix I that for the considered case of unitary  $Z$  noise, then a simple yet optimal strategy to estimate if a given rotation is above a certain threshold [task (i)] is to initialize the state  $\rho$  of the monitor qubits as  $|+\rangle^{\otimes N}$  and measure them in the  $X$  basis. An estimator  $\hat{\omega}$  for the true angle  $\omega$  can then be computed based on the number of times the “1” measurement outcome is observed. When  $\hat{\omega}$  is above a given threshold, one can assume that a defect occurred. Nonetheless, this procedure cannot distinguish between angles of opposite signs (rotations by  $\omega$  and  $-\omega$  render on average the same number of nontrivial measurement outcomes). Therefore, this scheme must be refined for task (ii), i.e., for the precise estimation of the rotation angle  $\omega \in [-\pi, \pi]$ . For this task, the strategy rather is to initialize the state  $\rho$  of the monitor qubits as  $|-\pi/4\rangle^{\otimes N}$ , where  $|-\pi/4\rangle := Rz(-\pi/4)|+\rangle$ , then measure half of them in the eigenbasis of the Pauli  $X$  operator and the other half in the  $Y$  operator. We devise an estimator  $\hat{\omega}$  in the appendix to optimally sense the rotation angle  $\pi$  which is detrimental to the present surface code construction (as it has no stabilizer signature). In both cases, we show in Appendix I that the monitoring strategy saturates the fundamental, asymptotic Cramér-Rao bound, and that it shows remarkable noise robustness. Namely, we demonstrate that even in the presence of dephasing and measurement noise, this theoretical bound is only marginally affected, by a factor  $(1 - \frac{5}{2}r)^{-1/2}$ , where  $r$  is the total noise strength. Given a sub-surface-code threshold noise  $r$  of the order of  $10^{-3}$ , the noisy output is only 0.1% larger than the noiseless one. In the following, we will assume  $r = 0.2\%$ .

*a. Task (i): Defect detection.* For task (i), we first define  $\omega_{\max}$  as the maximum value of  $\omega$  a logical snake can tolerate when it is shuttled near a defect, without inducing a dramatic increase of the logical error rate. We set  $\omega_{\max} = \omega_{\text{th}}/2$ , where  $\omega_{\text{th}} = 9\%$  is the error-correction threshold for such a scenario (assuming circuit-level noise of strength  $p = 0.1\%$  with one round of additional dephasing at a rate  $q = \sin^2(\omega/2)$ , see Appendix H for details). When  $\omega < \omega_{\max}$ , a strong exponential suppression of the error is guaranteed, and by an appropriate choice of the code distance  $d$ , the logical error rate can be kept low enough to run deep quantum algorithms. In contrast, if  $\omega \geq \omega_{\max}$ , the initially chosen code distance may fail to bring the error rate sufficiently low. Note that we here ignore ameliorations

provided by the complementary-gap scheme of Sec. III E. Besides, we set a detection threshold  $\hat{\omega}_{\max}$ , defined as the value of the estimator  $\hat{\omega}$  above which a defect is declared. This parameter controls the rate of false positives (deeming a nondefective shuttling link as defective) and false negatives (failing to detect a defect in a defective shuttling link). In our case, we want to minimize the latter, i.e., guarantee that almost every defect is detected, so as to minimize the logical error rate. This comes at the cost of a higher rate of false positives, meaning that a finite proportion of the shuttling links may be deemed defective even when they are not. While this has no impact on the logical error rate, it can lead to an increase in the computation time if the snake-surgery protocol of Sec. III C has to be used repeatedly. We will thus aim to keep this false positive rate within 5%–10%, while bringing the false negative rate as close to 0 as possible.

We illustrate the performance of the monitor qubits for task (i) in the left panel of Fig. 12. Namely, starting from a uniform distribution of the true angle  $\omega$  before monitoring, we plot its distribution  $p_{\text{mon}}(\omega)$  after the monitor-qubit-based filtering. The selection rule we choose is  $\hat{\omega} < \hat{\omega}_{\max} = \omega_{\max}/4$ , which leads to a rejection rate of the shuttling links of approximately 5% (see Appendix I). This amounts to invoking the reversion option in snake-surgery protocol approximately 5% of the time. The vertical dashed lines indicate the value of  $\pm\omega_{\max}$ . One must minimize the probability  $P_m$  that a qubit is shuttled through a postselected shuttling link whose true angle  $\omega$  is outside the window defined by the dashed lines. This is computed as

$$P_m = 2 \int_{\omega_{\max}}^{\pi} p_{\text{mon}}(\omega) d\omega. \quad (6)$$

For 900 monitor qubits, i.e., interlacing one monitor per data qubit in a  $30 \times 30$  surface code,  $P_m$  is as low as  $10^{-8}$ . More detailed plots can be found in Appendix I.

*b. Task (ii): Angle estimation.* For task (ii), we simply plot the RMS error  $\Delta\omega$  of the estimator  $\hat{\omega}$  with respect to  $\omega \in [-\pi, \pi]$ . The dashed lines represent the corresponding Cramér-Rao bounds as  $1/\sqrt{N}$  and confirm that even for a relatively low number of shots these asymptotic bounds are practically saturated near the rotation angles  $\pm\pi/2$ , 0 and  $\pm\pi$ . In other words, we are optimally sensitive to these rotation angles. Dominant noise contributions, i.e., dephasing and readout errors merely shift the dashed lines above by a fraction of a percentage. In contrast, in the worst case the estimator has a factor of  $\sqrt{2}$  larger RMS error (at angles  $\pm 3\pi/4, \pm\pi/4$ ) given we split the sensing states into two batches and use them to estimate different (but not independent) regions of the unit circle. In conclusion, using  $N = 800$  monitor qubits we can achieve an RMS

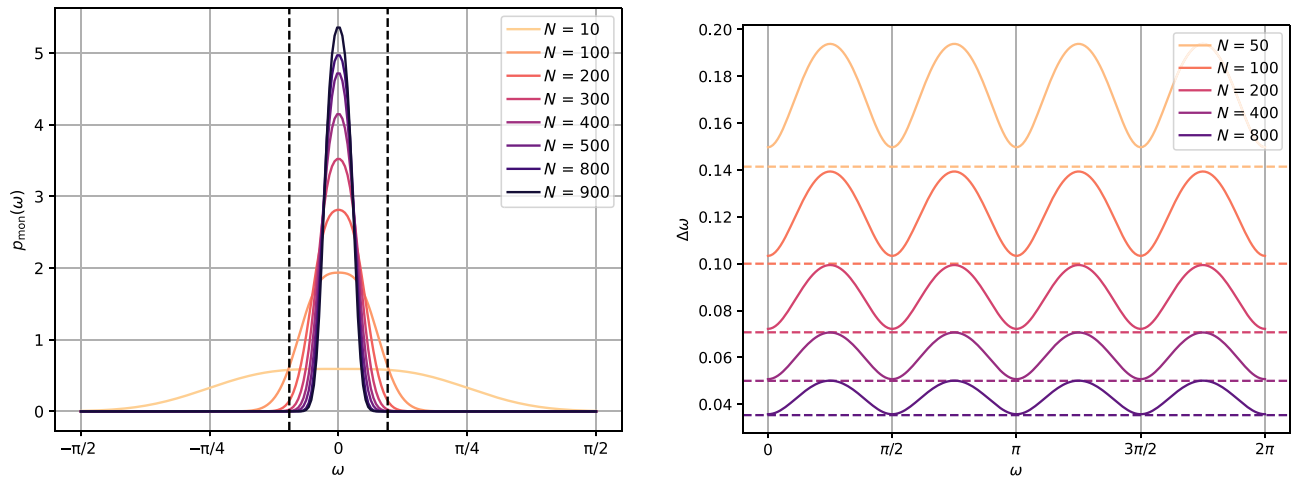


FIG. 12. Performance of the monitor qubits for the two tasks detailed in the main text.  $\omega$  and  $\hat{\omega}$  are respectively the rotation angle associated with the shuttling link, and its estimator computed from the measurement of  $N$  monitor qubits. The cumulative dephasing and measurement noise is set to  $r = 0.2\%$ . Left: performance of task (i), as the distribution of the angle  $\omega$ , postselected according to a selection rule leading to an approximately 5% rejection rate of the shuttling links. This amounts to canceling the shuttle via snake surgery (Sec. III C) approximately 5% of the time. The vertical dashed lines indicate the maximum angle  $\omega_{\text{max}}$  that the surface code can tolerate to guarantee fault tolerance (if not using the complementary-gap scheme of Sec. III E). Therefore, a logical error occurs when a qubit is shuttled through a postselected shuttling link, yet the rotation angle  $\omega$  is outside the window defined by the dashed lines. For  $N = 900$  monitor qubits, the probability of this occurring reaches  $10^{-8}$ . Right: performance of task (ii), as the RMS error  $\Delta\omega$  of the estimator  $\hat{\omega}$  with respect to  $\omega$ . The dashed lines represent the corresponding Cramér-Rao bounds as  $1/\sqrt{N}$  and confirm that even for a relatively low number of shots these asymptotic bounds are practically saturated near the rotation angles  $\pm\pi/2$ , 0 and  $\pm\pi$ , i.e., we are optimally sensitive to these rotation angles. Dominant noise contributions as dephasing and readout errors merely shift the dashed lines above by a fraction of a percentage.

error  $\Delta\omega \leq 0.05$ , which is one order of magnitude below the maximum tolerable angle  $\omega_{\text{max}} = 0.3$ .

The collected samples can also be split into batches of  $n$  bits in postprocessing to perform several independent estimations  $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_{N/n}$ . For instance, for  $n = 200$  each of these yields an uncertainty  $\Delta\omega \leq 0.1$  as illustrated in Fig. 12 (right). This is still comfortably below  $\omega_{\text{max}} = 0.3$ .

### E. Complementary gap

The previous section described a method capable of detecting defective shuttling links via the use of additional monitor qubits. Nonetheless, one must remember that throughout the course of the computation, information about potential errors is already collected by ancilla qubits measuring stabilizers. Therefore, in addition to enabling key logical error suppression, these measurement outcomes can be put to use for the assessment of the quality of a shuttling link. The general idea simply is that, when a snake is shuttled near a defect, its data qubits may experience a higher physical error rate, yielding a higher number of stabilizer events. As we suppose that defects are purely phaselike, we will solely focus on  $X$  stabilizers. In the present section, we formalize a criterion permitting us to distinguish with high probability between stabilizer signatures arising from the presence of a defect, or from normal

circuit noise, and to dramatically decrease the resulting logical error rate.

Note however that stabilizers alone cannot be utilized to detect the most catastrophic kind of defects, namely defects inducing a  $\pi$  rotation on all data qubits. These would indeed incur a logical failure with no stabilizer signature whatsoever (for odd-distance surface codes). Nevertheless, detecting such large rotations is the task where monitor qubits excel. Thus, one must think of the method presented in this section as a second filter used on top of the monitor-qubit-based scheme, the joint use of both protocols eventually leading to an almost-perfect detection of all defects.

Additionally, one can note that using this scheme inevitably leads to data qubits experiencing a high rate of errors due to the defect, which is only detected *a posteriori* through the stabilizer measurements. This is similar to the case of the previous section where monitor qubits are interlaced with data qubits. This defect detection could at first appear pointless if the data qubits have already undergone the defect-induced error channel. However, we demonstrated in Sec. III C that such errors can be reversed via so-called snake surgery, as long as they are phaselike and have been *a priori* identified.

The criterion we build for identifying defects from stabilizer measurements uses the recently introduced concept of *complementary gap* [44]. This quantity measures the

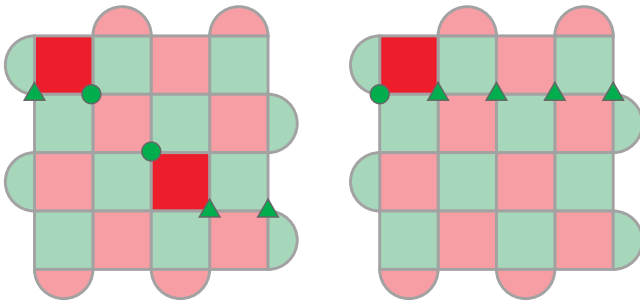


FIG. 13. Illustration of the complementary gap. Some  $X$  stabilizers indicate a nontrivial measurement outcome (bright red). In each case (left and right panels), this syndrome can be explained by various  $Z$ -error strings, two of which are represented with circles and triangles respectively. These specific error strings are the shortest-length error configurations that are consistent with the syndrome but yield opposite logical outcomes. The complementary gap is defined as their difference, i.e.,  $3 - 2 = 1$  and  $4 - 1 = 3$  in the left and right panel respectively. In both cases, the decoder would opt for the correction indicated by the circles; for the case on the right, the complementary-gap value is higher, meaning that the decoder is more confident in its decision.

confidence with which the decoder decided on a given correction. For instance, imagine a scenario where a single stabilizer at the center of a surface code renders a nontrivial measurement outcome. This syndrome can equally be explained by an error string terminating at either of the code’s boundaries. There is therefore a 50% chance of a logical failure. Since snake surgery grants us the effective ability to *reverse time* whenever a high-risk event is detected, it is in this case more wise to do so by measuring out the head of the shuttled snake.

More generally, we denote by complementary gap the length difference between the shortest error strings explaining the syndrome but yielding opposite logical outcomes. For instance, in the previous example of a single nontrivial stabilizer measurement right at the center of the code, the complementary gap is 0. More general cases are represented in Fig. 13.

As explained in Ref. [44], this quantity can be efficiently simulated by running the decoder with a switched boundary detector. Intuitively, when the physical error rate approaches the code’s threshold, e.g., due to a defect, the obtained complementary gaps typically decrease. If the other noise sources are kept sufficiently low below threshold, it is thus only natural to assert that a defect occurred when the observed complementary gap  $g$  is below a given value  $g_{\min}$ . As in the case of monitor qubits, our aim is to minimize the rate of false negatives while keeping the rate of false positives around 5%–10%. This guarantees that almost all defects are detected, while unnecessarily reverting the shuttle event in snake surgery 5%–10% of the time. In general, the value of  $g_{\min}$  permitting such a rejection rate depends on the code distance.

In Fig. 14 (left), we plot the distribution of postselected angles, assuming an initial uniform distribution (therefore not using additional monitor qubits here). The selection rule we use is  $g \geq g_{\min} = (d + 1)/2$ , which leads to the rejection of less than 5% of the shuttling links (see Appendix J). One can observe that the distribution is far less peaked than the one obtained from the sole use of monitor qubits. This means that the present method is *not* efficient at detecting defects. Rather, its strength lies in the significant logical error rate reductions it provides: even if a large defect angle  $\omega$  passes the filters, the action of rejecting instances with small complementary gaps highly improves the decoder’s performance, hence shrinking the rate of errors. This contrasts with the monitor-qubit case, for which the filtering has no such impact. Therefore, we plot in Fig. 14 (right) the logical error rate obtained after such postselection (circles) and compare it to the error rate that would be obtained if our detection scheme was not utilized (diamonds). The strength of this protocol is made evident, showing up to five orders of magnitude improvements for the plotted distances and error rates. Given the tendency observed in the plot, we expect even further reductions for higher code distances (see Appendix J).

### F. Defect resilience

Putting together the elements from all previous sections, one can estimate the resilience of our architecture to defects, as a function of the code distance  $d$  and the defect rate  $\rho$  per  $d$  stabilizer cycles. We additionally denote as  $P_L$  the normal logical error rate one would observe with a distance- $d$  rotated surface code under circuit-level noise at  $p = 0.1\%$  (with no defect), again over  $d$  stabilizer cycles.

When a snake is required to travel between two distant points in the device, its length is doubled and the head is detached from the tail which remains stabilized in place. The state of the shuttling path is then constantly monitored by the monitor qubits and stabilizer outcomes. If a defect is suspected, the head is destroyed and the shuttling event is restarted. We showed in Sec. III C that this procedure is fault tolerant, i.e., the snake does *not* pick up any error from the defect, as long they are purely phaselike (the error rate reduces to the normal logical error rate  $P_L$ ). Therefore, the suspected main source of error corresponds to the case where a defect occurs yet is left undetected. In this situation, one expects that the detection strategies designed in the previous sections would catch most of the defects (i.e., the distribution of postselected angles is peaked around the origin), and that the leftover angles can be handled by normal error correction, enhanced by the rejection of small complementary gaps.

We rigorously demonstrate in Appendix K that the final logical error rate  $\tilde{P}_L$  per  $d$  stabilizer cycles indeed reduces to the contribution of undetected defects, plus terms smaller than  $P_L$ . In particular, we show that, owing

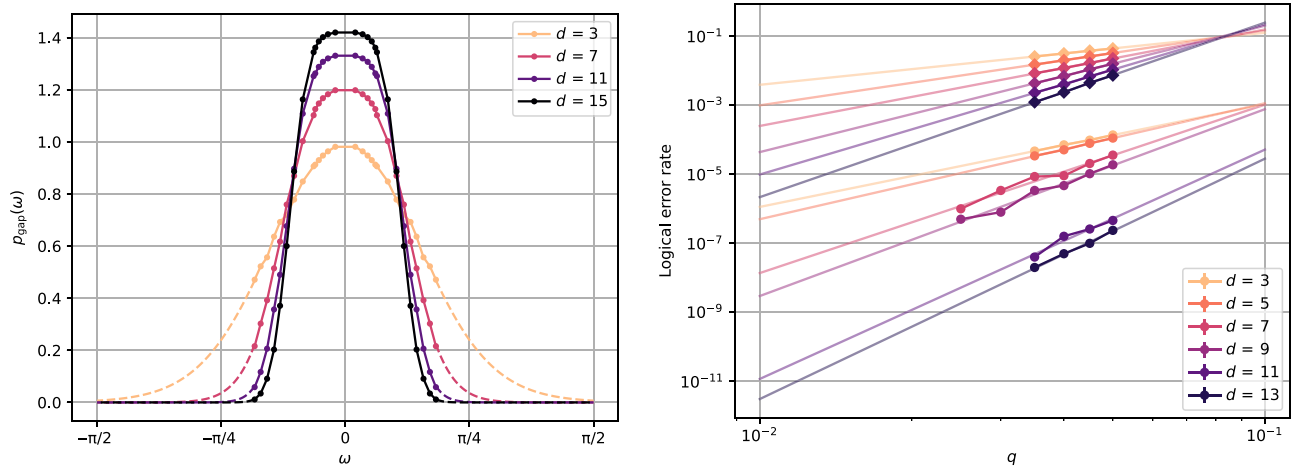


FIG. 14. Performance of the complementary-gap-based detection scheme for various code distances  $d$  under circuit-level noise at  $p = 0.1\%$  (see Appendix G), with one round of additional defect-induced dephasing at a rate  $q = \sin^2(\omega/2)$ . Left: distribution of postselected defect angles  $\omega$ , assuming a uniform initial distribution and according to a selection rule leading to the rejection of less than 5% of the shuttling links. Additional filtering from monitor qubits is not implemented here. The tails of the distributions (dashed lines) are obtained by Gaussian extrapolation. Right: logical error rate after postselection (circles), compared to the logical error rate without postselection (diamonds). Additional linear regressions (in log scale) are plotted for both datasets.

to the low rate of false positives that we set, the rejection of multiple shuttling paths until a nondefective one is found does not lead to any error accumulation. The defect-induced term in  $\tilde{P}_L$  is of the form

$$\rho \times \int_{|\omega|=0}^{\pi} P(\omega) p_{\text{both}}(\omega) d\omega, \quad (7)$$

where  $\rho$  is the rate of defects,  $P(\omega)$  is the probability of a logical error given that a defect characterized by an angle  $\omega$  is left undetected, and  $p_{\text{both}}(\omega)$  is the distribution of postselected angles from both defect-detection techniques.

As the monitor qubits are never entangled with the data or ancilla qubits of the error correcting codes, it follows that, for each value of  $\omega$ , the monitoring and complementary-gap strategies outcomes are uncorrelated. In other words, the detection strategies are conditionally independent given any defect angle  $\omega$ . Therefore one can write

$$p_{\text{both}}(\omega) = \frac{p_{\text{mon}}(\omega) p_{\text{gap}}(\omega)}{\int_{-\pi}^{\pi} p_{\text{mon}}(\omega) p_{\text{gap}}(\omega) d\omega}. \quad (8)$$

Consequently, all quantities involved in the calculation of the final logical error rate  $\tilde{P}_L$  have already been computed and analyzed:  $P(\omega)$  and  $p_{\text{gap}}(\omega)$  in Fig. 14, and  $p_{\text{mon}}(\omega)$  in Fig. 12 (left).

In Fig. 15, we thus plot the quantity  $P(\omega) p_{\text{both}}(\omega)$  for various code distances, using the data and extrapolations from the previous sections. This product represents the probability that a defect of angle  $\omega$  is undetected and triggers a logical error. We use one monitor qubit per data

qubit, i.e.,  $N = d^2$  monitor qubits.  $P(\omega)$  is a growing function of the defect angle  $\omega$  (higher angles yield higher error rates) while  $p_{\text{both}}(\omega)$  is decreasing with  $\omega$  (high angles are more likely to be detected by the monitoring schemes). Therefore, their product should feature a maximum, which is what we observe. Values of  $\omega$  closer to 0 are not plotted as the effect of the circuit-level noise would become predominant and the linear extrapolations we used from Fig. 14 (right) would break down.

Every code distance is associated with a given achievable logical error rate  $P_L$  under circuit-level noise at

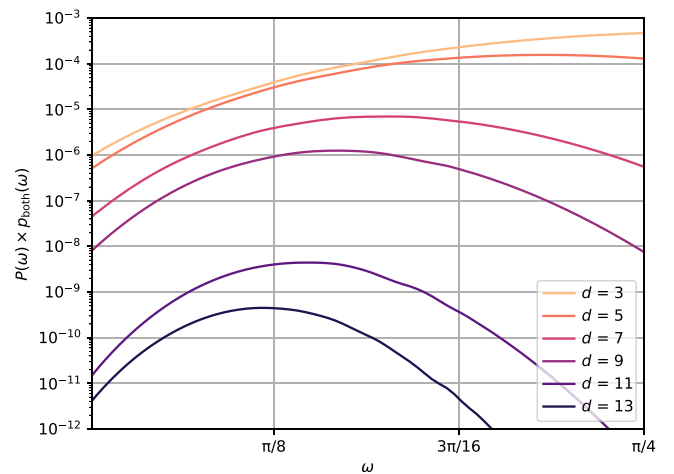


FIG. 15. Probability that a defect of angle  $\omega$  is undetected and triggers a logical error, plotted against  $\omega$  and for various code distances  $d$ . We use one monitor qubit per data qubit, i.e.,  $N = d^2$  monitor qubits.

$p = 0.1\%$  and in the absence of defects. By computing the integrals under the curves, one can observe that for all code distances  $d \leq 13$ , these are smaller than the normal error rates  $P_L$ ,

$$\int_{|\omega|=0}^{\pi} P(\omega)p_{\text{both}}(\omega)d\omega \leq P_L. \quad (9)$$

The ratio between the left- and right-hand side of the inequality also decreases with  $d$ , suggesting that it would be satisfied for *any* code distance. The integrals are computed by making use of the extrapolations of the previous sections. As noted before, the linear extrapolations used for  $P(\omega)$  are imperfect at small  $\omega$ , but this corresponds to minimal values of  $P(\omega)$  (orders of magnitude lower than at its maximum), hence to a good approximation this has no impact on the final value of the integral.

Evidently the combined use of our defect detection and recovering strategies is extremely powerful, actually bringing the impact of defects below the normal logical error rate. One can now wonder to what extent both methods are necessary. On the one hand, only using monitor qubits does not seem to be sufficient, as we observed in Eq. (6) that for a  $30 \times 30$  surface code using 900 monitor qubits, the probability of observing an angle  $\omega$  leading to a logical error was  $P_m = 10^{-8}$ . If one aimed to reach error rates as low as  $10^{-15}$ , a defect rate  $\rho = 10^{-7}$  would thus be necessary to bring defect-induced noise as low as the normal logical error rate. This does not mean that monitor qubits must be overlooked however, as they are the only way to sense angles around  $\pi$ , which stabilizers cannot detect. On the other hand, the postselection of high complementary gaps promises massive error rate reductions, with a post-selected error rate  $P(\omega)$  orders of magnitude lower than the normal error rate  $P_L$ . Similarly to Eq. (9), one can show that

$$\int_{|\omega|=0}^{\pi/2} P(\omega)p_{\text{gap}}(\omega)d\omega \leq P_L. \quad (10)$$

This means that, if angles closer to  $\pi$  are excluded, the complementary gap method is sufficient to bring the defect-induced error rate below the normal logical error rate. Therefore, while both detection strategies appear necessary, one improvement one could think of is to reduce the number of monitor qubits so as to only reject angles around  $\pi$ . In all other cases, stabilizers aided by the postselection of high complementary gaps would be able to lower the defect-induced error rate below the normal logical error rate. Adaptive measurement strategies could also be considered to even further reduce the overhead requirement induced by monitor qubits—measuring by batches and only deploying the full monitor qubits suit when initial measurements do not provide enough confidence.

Do note however that in all our derivations, we did assume a low rate of defects, such that two defects would

not occur during the same shuttling event, and such that defect declarations would mostly arise from false positive detections (at a rate of 5%–10%). This is typically true if  $\rho$  is much smaller than 5%–10%, i.e., smaller than 1 defect every  $10d$  stabilizer cycles on a given shuttling link.

#### IV. UNIVERSAL QUANTUM COMPUTATION

In this section we demonstrate that universal quantum computation is possible on the snakes on a plane architecture, and give protocols for the implementation of a universal set of gates.

##### A. Logical-level connectivity

One of the strengths of our proposal is that it boasts an all-to-all connectivity at the logical level. This is permitted by the expected fast and high-fidelity shuttling of silicon-spin qubits, enabling the global movement of the snakelike logical qubits across the device. During these long shuttles, the preservation of a low logical error rate is guaranteed by the protocols of Sec. III. In particular, it relies on the rejection of a finite number of shuttles, whenever an abnormal event is detected by monitor qubits or stabilizer measurements. If such an abnormality is repeatedly detected, a defect can be declared, ultimately leading to the deactivation of the shuttling link. Additionally, some  $2 \times N$  filaments may be deemed defective even before the computation started, as the presence of fabrication defects will be inevitable in such large devices. Therefore, a finite proportion of the shuttling links will necessarily be unusable. Yet, we demonstrate in the following that logical-level all-to-all connectivity is still guaranteed as long as this proportion is below a given threshold.

In Fig. 4, snakes live in white tiles, which are connected through interaction edges (small edges where two white tiles are adjacent). Therefore, one can identify such tiles as the vertices of the logical-level connectivity graph, and the interaction edges as the edges. These form a (rotated) square lattice. In the worst case, one can assume that a defective shuttling link would lead to the entire deactivation of a white tile (vertex) or of an interaction edge (edge). In these cases, the maximum proportion of deactivated links that is tolerable to keep a fully connected lattice (at the scale of the entire device) is given by the site and bond percolation thresholds, which for the square lattice are respectively about 60% and 50%. Therefore, as long as the proportion of defective shuttling links is below 50%, all-to-all connectivity is guaranteed. However, the computation time may significantly increase when the number of disabled links approaches the percolation threshold. Indeed, the route a snake must follow to travel between two given points in the device may naturally be extended to avoid defective links.

## B. Universal set of gates

The set of universal gates we advocate for is the classic set {CNOT,  $H$ ,  $S$ ,  $T$ }. One of the strengths of our architecture is that transversal gates can seamlessly be implemented via the interaction edges, where two logical snakes can be made adjacent. CNOT gates can therefore be applied particularly efficiently, much faster than in more conventional static 2D setups where lattice surgery must be used.

As for Hadamard gates, they can be applied transversally as well, although this induces a 90-degree rotation of the surface code. This rotation can be reversed by physically reshuffling the qubits through the junctions. This however incurs a relatively high time overhead and changes the orientation of the logical operators of the code, meaning  $Z$  logical operators would now be parallel to the direction of shuttling, drastically reducing the protection against  $Z$  noise. While there may exist alternative architectures or protocols that permit an efficient and safe implementation of such rotations, we leave this investigation for further research. Alternatively, one possible implementation of Hadamard gates via domain wall patch deformation is given in Ref. [45]. It requires  $O(d)$  rounds of error correction and always leaves the  $Z$  logical operator orthogonal to the direction of shuttling.

As for  $S$  and  $T$  gates, we adopt the conventional choice of an implementation via gate teleportation and additional magic state distillation. This process relies on the concatenation of the surface code with an additional code whose  $S$  or  $T$  gate is transversal. The entire procedure is based on the successive implementation of many CNOT gates, which as explained before are not a bottleneck in our platform. One could even envision implementing the multitarget CNOTs appearing in the distillation circuits via quantum fan-out (as explained at the end of Sec. III C), which may greatly simplify them.

## C. Semitransversal CNOT gates

In order to apply a logical CNOT gate transversally, two logical qubits have to be entirely shuttled through an interaction edge, which amounts to a total shuttling distance of  $O(d^2)$ . If one were to implement all transversal physical gates at once, this would imply  $O(d^2)$  shuttling increments without stabilizer measurements, while these are normally performed in  $O(d)$  shuttling increments (Figs. 3 and 5). This scaling becomes problematic when shuttling operations are noisy, leading to greater losses in fidelity.

To address this problem, we can employ a semitransversal gate protocol (Fig. 16). The fundamental principle behind semitransversal gates involves performing transversal operations in discrete batches while concurrently stabilizing all physical qubits that are not actively involved in the interaction. By batching the transversal operations, it becomes possible to reduce the cumulative impact of shuttling noise. The number of transversal gates

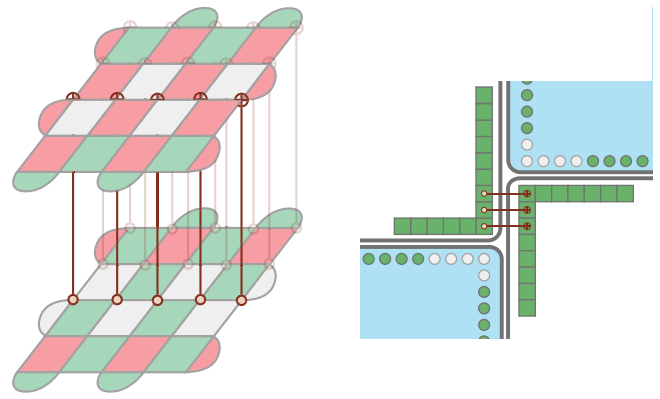


FIG. 16. Semitransversal CNOT gate. In this protocol, a logical CNOT is implemented by applying transversal CNOTs in batches, e.g., row by row. CNOTs that are applied at the current round are represented in brighter brown, while CNOTs that have previously been applied are drawn in a faded brown. Some stabilizers cannot be measured at the current round due to their anticommutation with certain CNOTs: these are drawn in light gray. Left panel: canonical 2D representation of a surface code. Right panel: snakes on a plane representation.

executed per batch can be adjusted based on the shuttling noise levels. Specifically, larger batches yield faster but noisier logical gates (the limiting case being the implementation of a normal transversal gate, with one batch of  $d^2$  CNOTs). The length of the interaction edge can basically be set to the number of transversal operations per batch.

Because transversal CNOTs are not applied all at once, some  $X$  (resp.  $Z$  stabilizers) have a support that only partially overlaps with control (resp. target) qubits of already-implemented CNOTs. These cannot be measured as they would lead to a propagation of the errors. More precisely, let us denote as  $|\Psi\rangle$  the state of the two-qubit logical state before applying the CNOTs. By definition, the application of any stabilizer  $S$  should leave  $|\Psi\rangle$  unchanged,

$$S|\Psi\rangle = |\Psi\rangle. \quad (11)$$

When applying a unitary operation  $U$  on  $|\Psi\rangle$ , only stabilizers that leave  $U|\Psi\rangle$  unchanged can be measured; otherwise the state could randomly be projected outside the logical subspace,

$$SU|\Psi\rangle = U|\Psi\rangle. \quad (12)$$

Now let us assume that  $S$  is an  $X$  stabilizer acting on four data qubits, indexed from 1 to 4, in the control logical qubit. Let us additionally assume that transversal CNOTs have only been applied between qubits 1 and 1', and 2 and 2' (where the  $i'$  denotes qubit indices in the target logical

qubit). It follows:

$$\begin{aligned} SU|\Psi\rangle &= X_1X_2X_3X_4\text{CNOT}_{1,1'}\text{CNOT}_{2,2'}|\Psi\rangle \\ &= USX'_1X'_2|\Psi\rangle \\ &= UX'_1X'_2|\Psi\rangle, \end{aligned}$$

where the equality between the first and second lines follows from CNOT and  $X$  gates commutation rules.  $X'_1X'_2$  however is not a stabilizer of the target logical qubit ( $X'_1X'_2X'_3X'_4$  is). Therefore, Eq. (12) is not respected. With this logic, one can easily deduce that, at all times, all  $X$  stabilizers on the target logical qubit and all  $Z$  stabilizers on the control logical qubits can be measured. However,  $X$  (resp.  $Z$ ) stabilizers whose support only partially overlaps with already-implemented CNOTs on the control (resp. target) logical qubits cannot be measured. This is represented in Fig. 16.

## V. DISCUSSION

In this paper, we explore the practicality of shuttling logical qubits in a latticework structure. The structure is designed to be suitable for realization with silicon-spin chip devices. Our approach leverages ideas presented in the  $2 \times N$  architecture of Ref. [18]. In that earlier piece of research, it was demonstrated that certain code classes could efficiently be embedded on a  $2 \times N$  array equipped with shuttling, and that universal quantum computation was possible. However, due to the linear structure of the device, the connectivity at the logical level was overly constrained, leading to relatively long run times. Additionally, single-point failures would cause the device to be split into two disconnected halves.

Here we have demonstrated that interfacing a large number of such structures together through junctions would solve the aforementioned problems, while preserving a low-density device. The proposed latticework arrangement employs linked  $2 \times N$  filaments along which logical qubits can travel. Contrary to our original  $2 \times N$  paper [18], logic is scalable in this new proposal. Indeed, it permits long-range shuttling of logical qubits and indeed large-scale fault-tolerant quantum computation through semitransversal gate operations. Besides, while the length of the snakelike logical qubits now scales as  $d^2$ , this only translates into lengths of at most 100  $\mu\text{m}$  for practical values (considering a logical qubit composed of 1000 physical qubits and an interdot distance of 100 nm). For demonstrated shuttling speeds between 10 and 100 m/s, this would mean only a few microseconds to shuttle an entire snake. Subsequent interactions would then be implemented transversally, which represents a minimal time overhead compared to more conventional lattice-surgery procedures. Therefore, the total time taken by our logical gates will not be prohibitively large, even in mature fault-tolerant devices.

These conclusions are supported by a detailed analysis of the noise processes at hand and the design of bespoke protocols to minimize their impact. We first characterize the noise experienced by silicon-spin qubits in our device, in the form of charge noise. We demonstrate that this error source can be ameliorated by the right choice of qubit encoding, namely a singlet-triplet encoding. As such a construction lives in a decoherence-free subspace, it is naturally first-order immune against phase errors. However, the sole use of this encoding would not be sufficient to protect the device against the most catastrophic noise source: the appearance of pin defects materializing near a shuttling track. This could incur a dramatic error surge yielding a logical failure. We therefore present a protocol that actively addresses the issue and provides a profound suppression of logical errors. The method involves the high-fidelity detection of defects, via the analysis of the collected stabilizer measurements and the use of additional monitor qubits. Armed with this capability, we show that one can revert a process that is suspected to have involved a defect-induced error. In particular, assuming that the defect-induced error mechanism is purely  $Z$ -like, we demonstrate that the errors it incurs can perfectly be annihilated, by effectively *reversing time* and going back to the pre-defect version of the erroneous snake. Jointly using all above protocols, we demonstrate a tolerance to defects exceeding the tolerance to normal circuit-level noise, and for any code distance. In the presence of transverse magnetic field gradients however, it may be necessary to consider defect-induced  $X$  flips on top of the studied case of  $Z$  flips. If these  $X$  errors are sufficiently small and noncorrelated, normal error correction would be able to tackle them. Otherwise, one could resort to the use of a lower- or higher-level code to catch and correct rare but disruptive bit-flip like events. Alternatively, by using the dual-spin encoding we advocate for in Sec. III B, we expect that correlated  $Z$  rotations between multiple pairs of qubits are practically impossible. Indeed, if both physical qubits in an odd-parity pair are affected by the same  $Z$  rotation, due to shuttling in the vicinity of a defect, then their state is unchanged (only a global phase is applied). Therefore, snake surgery could be repurposed for the correction of correlated  $X$  flips instead, provided they are even observed. We leave this investigation to further research.

In addition to these noise-resistance schemes, we show that our proposed architecture is suited for universal quantum computation. Its first strength is the all-to-all connectivity at the logical level, enabled by long-distance shuttling. In particular, we note that a high proportion of defective edges is tolerable without the loss of connectivity. The second strength of our current proposal is the possibility to interface logical qubits so as to perform transversal gates, which represents a considerable time reduction compared to more common lattice-surgery protocols. In this context we describe a generalized *semitransversal*

implementation of two-qubit logical gates. This scheme offers an opportunity to implement transversal gates in batches rather than all at once (which would be the case in a traditional transversal setup).

We conclude that the use of monitored, fault-tolerant snakelike logical qubits (“MF snake qubits”) is a natural solution to realizing scalable quantum computing in two-dimensional solid-state architectures that support shuttling. The latticework structure on which the MF snakes move has a high degree of design flexibility, and its locally linear nature is compatible with near-term silicon manufacturing capabilities.

The natural next step would be the design of protocols translating the low-level features of the SoaP architecture all the way up the stack to the algorithm level. While it is possible to take a standard surface code approach and map it to SoaP (provided fast and high-fidelity shuttling), our architecture has many properties that can lead to significant time reductions. These include our novel two-qubit gate implementation strategy and an all-to-all connectivity. In future research, we therefore plan to study how these elements can be combined to optimally implement  $T$  gates, compile an algorithm on our architecture or route the snakes in the presence or absence of defects.

We based this proposal on surface code quantum error correction (QEC)—while this code suffers from a high qubit overhead, silicon-spin qubits footprint is so small that this choice should not be problematic even in the utility-scale regime. However, as explored in our previous work [18], many classes of quantum low-density parity check (qLDPC) codes are a good fit for the  $2 \times N$  qubit layout and therefore also for our architecture. Such codes can in fact naturally be implemented by accepting an increased shuttling distance or number of shuttles within each stabilizer cycle. The addition of junctions in the SoaP architecture offers even higher opportunities, by allowing one to reshuffle qubits within a snake (see Fig. 17)—this was not possible in our previous work. Hence, one can further optimize the syndrome extraction circuits of the

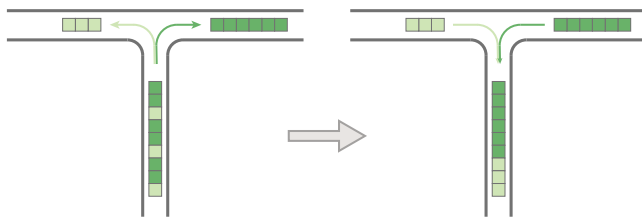


FIG. 17. Use of a junction to reshuffle the physical qubits within a snake. This procedure can be utilized to implement a wider class of high-rate qLDPC codes more efficiently. Left panel: light green data qubits are sent to the left and dark green data qubits to the right. Right panel: when all qubits have gone through the junction, the snake is reassembled by first shuttling back all light green qubits, then all dark green qubits.

proposed codes, or even find novel classes of codes whose efficient implementation is enabled by the use of junctions. Since all of the mentioned classes of codes in Ref. [18] are Calderbank–Shor–Steane (CSS) codes, they all have a transversal CNOT gate that can be implemented in a similar semitransversal fashion as proposed in Sec. IV C. The same requirement holds—one needs to be careful not to measure stabilizers for which only a partial set of data qubits in their support have undergone the semitransversal operation. Note that performing a transversal CNOT between higher-rate codes implements a logical CNOT on each pair of respective logical qubits and generally that may be undesirable.

We leave the adaption of snake-surgery protocol for higher-rate codes for future work.

## ACKNOWLEDGMENTS

The authors are grateful to Zhu Sun for helpful conversations. The numerical modeling involved in this study made use of the Quantum Exact Simulation Toolkit (QuEST), and the recent development pyQuEST which permits the user to use Python as the interface front end. We are grateful to those who have contributed to these valuable tools. The authors acknowledge support from the EPSRC QCS Hub Grant (Agreement No. EP/T001062/1), EPSRC’s Robust and Reliable Quantum Computing (RoarQ) Project (No. EP/W032635/1), and the Project (No. EP/Y004655/1). B.K. thanks UKRI for the Future Leaders Fellowship project titled Theory to Enable Practical Quantum Advantage (MR/Y015843/1).

## DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

## APPENDIX A: CHOICE OF A PLATFORM: SILICON SPINS

One qubit platform is particularly well suited to the implementation of all the operations described above: silicon-spin qubits. While we summarize key points below, nonexpert readers can refer to these excellent reviews for further explanations: Refs. [46,47].

From a general point of view, silicon-spin qubits appear as an excellent candidate for the implementation of a quantum computer, due to their high coherence times [15], scalability [46], compatibility with advanced manufacturing techniques [48,49] and cryogenic classical electronics [50,51]. Small silicon-spin qubit processors have already been engineered [52] and small error correcting codes

have been implemented [53]. Single- and two-qubit gates with fidelity above 99% have been demonstrated [10–12] even in commercially sourced chips [13] and fidelities beyond 99.9% have been reported [14]). Among the implementable two-qubit gates, we find the CZ and CNOT gates, which are both natural operations for stabilizer measurements.

More specifically to our proposed architecture,  $2 \times N$  arrays of silicon-spin qubits have already been demonstrated [54]. Moreover, recent experiments succeeded in shuttling electrons across a silicon-spin qubit chip with 99.99% fidelity per shuttling increment and at a high speed [9]. This rapid development compared to previously demonstrated shuttling capabilities [7,8] promises to bring even better fidelities in the near future. We are therefore confident that silicon-spin qubits are a promising candidate for the implementation of the shuttling-based fault-tolerant protocol we advocate for in this piece of research [47].

We will base all numerical parameters on state-of-the-art silicon-spin qubit technologies. Namely, while two-qubit gates require around  $0.1 \mu\text{s}$  [10–12,55]), initialization, measurement and single-qubit gates can last up to a few microseconds [52,56,57]. The modes of shuttling described in Fig. 5 require  $2d$  and  $d$  shuttling increments respectively. Supposing  $d \sim 30$  for relevant applications [2], an interdot distance  $l_{dd} = 100 \text{ nm}$  and a shuttling speed  $v = 10\text{m/s}$  [39], shuttling would take up 300–600 ns in a stabilizer cycle. Adding this contribution to the initialization, gates and measurement times, one can realistically expect a stabilizer cycle time of  $3 \mu\text{s}$ .

### APPENDIX B: PRIOR LATTICEWORK SCHEME: THE LOOPED PIPELINE ARCHITECTURE

The layout supporting the Snakes on a Plane architecture is a latticework structure of one-dimensional arrays. This is not the first instance of such a latticework-based layout: previous research conducted by authors of the present paper led to the Looped Pipeline.

In this proposal, a logical qubit is spread thinly, so to speak, over the latticework; each local region (a square-shaped loop in Fig. 18) hosts a single data qubit or ancilla qubit of a given logical qubit. The highlighted blue circles correspond to four data qubits and one ancilla; we observe that as the physical qubits move according to the arrows, the ancilla will encounter each of the data qubits in turn, thus permitting a stabilizer measurement via measurement of the ancilla, using devices in the void spaces (not shown). As Ref. [20] explains in detail, by hosting multiple physical qubits in each loop the architecture can effectively *stack* logical qubits. This means that it is equivalent to a 3D lattice where  $k$  surface codes would be sitting on top of each other, where  $k$  is the number of qubits per loop (see Fig. 4 of Ref. [20]). Charge defects are not as worrisome in this

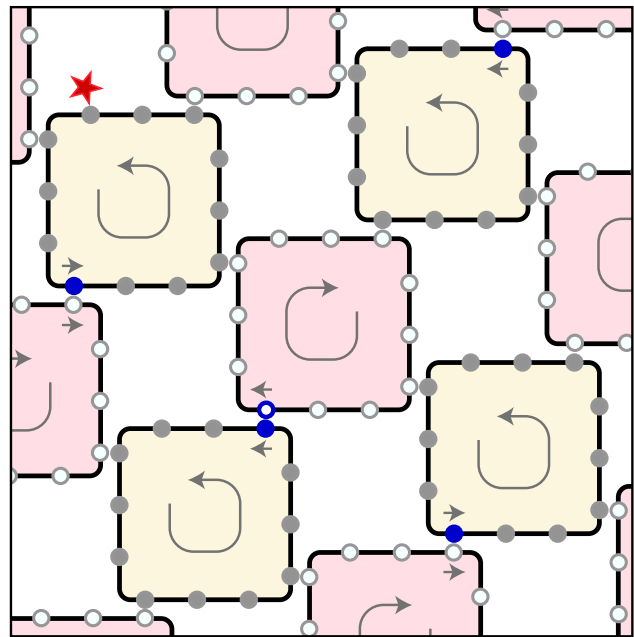


FIG. 18. The Looped Pipeline architecture. Contrary to the Snakes on a Plane architecture, physical qubits belonging to the same logical qubit are spread across the device. Ancilla qubits (hollow circles) go around the pink loops and periodically meet the data qubits (filled circles) it needs to interact with. One such example of ancilla and data qubits is highlighted in blue. When  $k > 1$  data and ancilla qubits are embedded in each loop (here  $k = 12$ ), a 3D lattice is virtually generated, containing  $k$  logical qubits are effectively *stacked* on top of each other. The red star indicates the presence of a defect in the lattice: contrary to SoaP shuttling near a defect does not scratch an entire logical qubit but rather affects a single data qubit in all the surface code patches, making it easier to handle by standard error-correction protocols.

architecture as in SoaP as they would only affect a single data qubit in each logical qubit (red star in the figure).

### APPENDIX C: ALTERNATIVE LATTICEWORK STRUCTURES

We here give two examples of arrangements of the  $2 \times N$  devices, other than the one studied in the main text (Fig. 4). These may allow for different connectivity opportunities at the logical level. Figure 19 represents a hexagonal lattice, using three-way junctions only as in the main text. Figure 20 is a rectangular lattice, made of four-way junctions.

### APPENDIX D: MODELING THE FLUCTUATING ENVIRONMENT

The authors of Ref. [38] represent the fluctuating field  $B(x, t)$  as a zero-mean Ornstein-Uhlenbeck sheet. As it is a Gaussian random sheet with a defined mean, its behavior

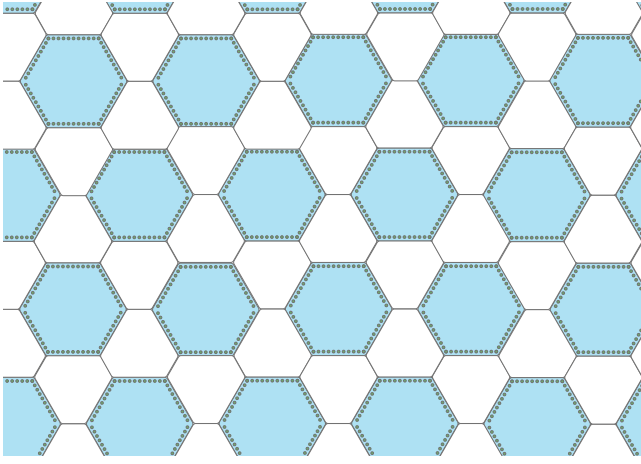


FIG. 19. Hexagonal lattice using three-way junctions.

is fully determined by its covariance matrix,

$$C((x_1, t_1), (x_2, t_2)) = \frac{\sigma^2}{4\kappa_x\kappa_t} e^{-\kappa_x|x_1-x_2|-\kappa_t|t_1-t_2|}, \quad (\text{D1})$$

where  $\sigma$  is the process' standard deviation, and  $\kappa_x = 1/\lambda$ ,  $\kappa_t = 1/\tau$  are respectively the inverse correlation length and inverse correlation time of the fluctuating magnetic field. Note that we suppose that the field only depends on a single spatial dimension corresponding to the shuttling path. Here, we set  $\lambda = 100$  nm,  $\tau = 20$   $\mu$ s and  $\sigma^2/4\kappa_x\kappa_t = \sqrt{2}\kappa_t$ . These parameters are extracted from Ref. [38] and corresponds to typical values found in experiments for Si/SiGe devices.

While shuttling through such an environment, a system of  $N$  noninteracting qubits will experience the following Hamiltonian [38],

$$H(t) = \frac{g_0\mu_B\hbar}{2} \sum_{i=1}^N B(x_i(t), t)\sigma_z, \quad (\text{D2})$$

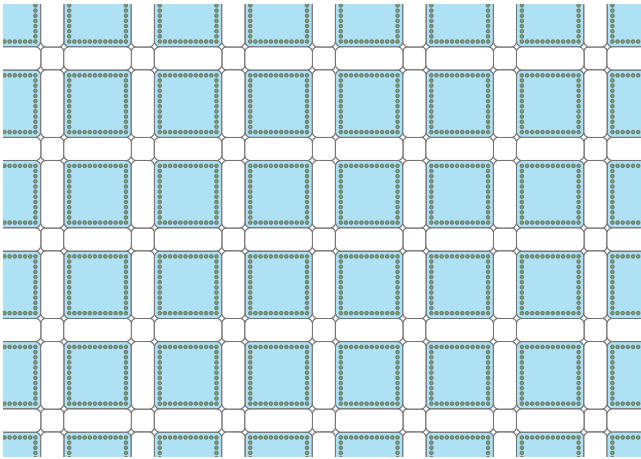


FIG. 20. Rectangular lattice using four-way junctions.

where  $g_0$  is an averaged  $g$ -factor,  $\mu_B$  is the Bohr magneton,  $x_i(t)$  is the time-dependent position of the center of the wave packet describing the  $i^{\text{th}}$  electron, and  $\sigma_z$  is the Pauli  $Z$  matrix.

To compare different encodings, we can focus on characterizing the random phase acquired when the qubits are shuttled. For simplicity, consider a single qubit shuttled for a time  $T$ . The phase it will acquire under the Hamiltonian described in Eq. (D2) is given by

$$\phi(T) = \frac{g_0\mu_B\hbar}{2} \int_0^T B(x_1(t), t)dt. \quad (\text{D3})$$

The density matrix of the electron at time  $T$  thus reads

$$\rho(T) = \begin{pmatrix} \rho_{00}(0) & \rho_{01}(0) \exp(-i\phi(T)) \\ \rho_{10}(0) \exp(i\phi(T)) & \rho_{11}(0) \end{pmatrix} \quad (\text{D4})$$

with  $\rho_{ij}(0)$  representing the coefficients of the initial density matrix. Note that only the off-diagonal elements are affected by the shuttling channel. This phase will vary between experiments and could lead to the complete dephasing of the state when averaging the results. The quantity of interest is thus the dephasing factor  $W(T)$  obtained by averaging the phase  $\exp(-i\phi(T))$  over the random sheet  $B$ . This factor appears in the averaged density matrix as follows:

$$\mathbb{E}[\rho(T)] = \begin{pmatrix} \rho_{00}(0) & \rho_{01}(0)W(T) \\ \rho_{10}(0)W^*(T) & \rho_{11}(0) \end{pmatrix}, \quad (\text{D5})$$

where  $W(T) = \mathbb{E}[\exp(-i\phi(T))]$ . Reference [38] generalizes this derivation to more than one electron and gives analytic expressions for  $W$  for both the LD and ST encodings.

Their model does have limitations; it does not directly represent charge noise as it does not have a  $1/f$  spectrum and it does not account for the valley degree of freedom. Nevertheless, since the main point of this Appendix is to convey the idea that the choice of encoding can drastically reduce the effect of a fluctuating magnetic field, and as the valley splitting is believed to be large in Si/SiO<sub>2</sub> devices, we will also adopt this model. It allows us to account for slow variations of the  $g$ -factor landscape while being simple enough to analytically derive the impact of the environment on shuttled qubits. We leave the inclusion of charge noise and the valley degree of freedom for future work.

## APPENDIX E: SHUTTLING INFIDELITY FOR DIFFERENT ENCODINGS

In Fig. 13, we plotted the evolution of the infidelity with respect to speed for different values of the distance

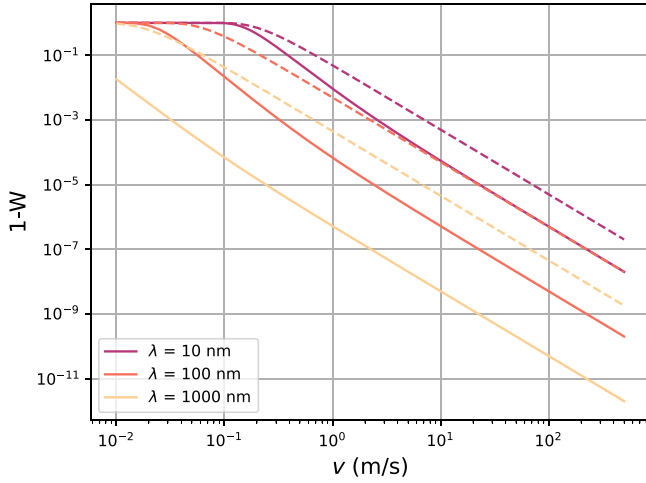


FIG. 21. Evolution of the shuttling infidelity as a function of speed for different values of the correlation length  $\lambda$  with  $d = 100$  nm and  $\tau = 20$   $\mu$ s. The solid and dashed lines correspond to the ST and LD encodings respectively.

between the ST qubits. In order to confirm our understanding, we study here the evolution of the shuttling infidelity for different values of the correlation length  $\lambda$  in Fig. 21 and correlation time  $\tau$  and Fig. 22. We find that the ST encoding outperforms the LD encoding and that the infidelity decreases with the shuttling speed, which has been explained in the main text. In addition, we find that as  $\lambda$  or  $\tau$  increases, the infidelity decreases. This is consistent with the fact that by increasing these quantities we reduce the spatial and time fluctuations, ultimately creating a smoother shuttling track for both encodings. More precisely, by increasing  $\lambda$ , the protection power of the ST encoding in Fig. 21 is improved for any speed  $v$ , as seen

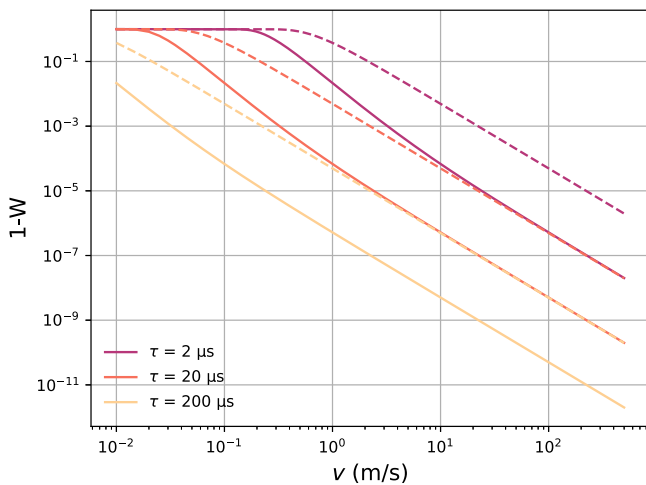


FIG. 22. Evolution of the shuttling infidelity as a function of speed for different values of the correlation time  $\tau$  with  $d = \lambda = 100$  nm. The solid and dashed lines correspond to the ST and LD encodings respectively.

by the growing difference in infidelity between solid and dashed lines of the same color. This comes from the fact that the difference in experienced magnetic field between the two electrons becomes smaller as the correlation length increases. As for Fig. 22, the protection is only increased up to a certain speed due to the fact that the delay between the electrons decreases with  $v$ . After this threshold the fidelity improvement is limited by the correlation length.

## APPENDIX F: SNAKE SURGERY: RECOVERING FROM DRAMATIC PHASE ERRORS AFTER TRANSVERSAL GATES

We here show that the protocol described in Sec. III C can also be applied when the logical snakes are interacted *before* a defective shuttling link was detected.

In order to bring two logical snakes closer together such that they can interact, they are first grown then split in halves, as explained in Sec. III C. This renders the states,

$$|\psi\rangle = \alpha |++\rangle + \beta |--\rangle, \quad (\text{F1})$$

$$|\psi'\rangle = \alpha' |++\rangle + \beta' |--\rangle, \quad (\text{F2})$$

where  $|\psi\rangle$  and  $|\psi'\rangle$  are the snakes states before the growth and split. The heads are then shuttled while the tails are stabilized in place (which is protected against scratches as stabilizer cycles only requires  $O(d)$  shuttling increments).

The application of transversal CNOTs to the heads brings the system to the following state:

$$|\Psi\rangle = (\alpha |+\rangle + \beta |-\rangle) |0\rangle (\alpha |++\rangle + \beta |--\rangle) \\ + (\alpha |+\rangle - \beta |-\rangle) |1\rangle (\alpha |++\rangle - \beta |--\rangle),$$

where the four qubits are respectively the tail and head of the first logical qubit, then the tail and head of the second logical qubit.

One first case corresponds to a successful interaction, where no defect was detected during the shuttle. In this case the tails are measured in the  $Z$  basis, which projects  $|\Psi\rangle$  onto

$$|\Psi\rangle \rightarrow \frac{1}{2} [(\alpha + (-1)^{m_1} \beta) |0\rangle (\alpha' |+\rangle + (-1)^{m_2} \beta' |-\rangle) \\ + (\alpha - (-1)^{m_1} \beta) |1\rangle (\alpha' |+\rangle - (-1)^{m_2} \beta' |-\rangle)],$$

where  $m_1$  and  $m_2$  are the measurement outcomes of the tail measurements. By applying Pauli corrections in the form  $X_1^{m_1} X_2^{m_2}$ , one can obtain the desired final state, i.e.,

$$|\Psi\rangle \rightarrow \text{CNOT}_{12} [(\alpha |+\rangle + \beta |-\rangle) (\alpha' |+\rangle + \beta' |-\rangle)]. \quad (\text{F3})$$

Now let us assume that the interaction was deemed unsuccessful due to the appearance of a defect during the shuttle. This translates into catastrophic  $Z$  errors affecting

the heads. In this case, they are measured in the  $Z$  basis, rendering the state,

$$|\Psi\rangle \rightarrow (\alpha|+\rangle + (-1)^{m_1}\beta|-\rangle)(\alpha'|+\rangle + (-1)^{m_1+m_2}\beta'|-\rangle).$$

By this time applying Pauli corrections in the form  $X_1^{m_1}X_2^{m_1+m_2}$ , one can recover the initial two-logical-qubit state,

$$|\Psi\rangle \rightarrow (\alpha|+\rangle + \beta|-\rangle)(\alpha'|+\rangle + \beta'|-\rangle). \quad (\text{F4})$$

## APPENDIX G: QEC SIMULATIONS DETAILS

In all QEC simulations in this paper, we only focus on  $Z$  errors as they are the main error mechanism at play in our system. This is possible since the surface code is a CSS code, hence  $X$  and  $Z$  errors can be analyzed separately. Therefore, we evaluate the  $X$  logical error rate of a distance- $d$  surface code over  $d$  rounds of stabilizer measurements, via  $N_{\text{runs}}$  runs of Monte Carlo simulations.  $N_{\text{runs}}$  ranges between 10 000 and 1 000 000 depending on the target logical error rates.

In all simulations, circuit-level noise of strength  $p$  is adopted, i.e., depolarizing channels after every initialization and single- or two-qubit gate, as well as a classical flip of every measurement outcome, all of strength  $p$ . Assuming high-fidelity shuttling, we showed in Ref. [18] that shuttling noise had a minimal impact on logical error rates compared to gate noise, thus we neglect it here. Additional noise processes described in further details in the relevant sections may be added as well.

The syndrome these errors create is decoded via Minimum Weight Perfect Matching [1]. The initial errors and the correction are then added up to determine if the  $X$  logical operator value was flipped.

## APPENDIX H: SURFACE CODE THRESHOLD IN THE PRESENCE OF A DEFECT

When the entire length of a snake is shuttled near a defect (i.e., for a long shuttle), all data qubits experience *one round* of increased error. More specifically, we assume that this defect-induced noise is pure  $Z$ , and for convenience of QEC simulations, we suppose that it is a dephasing channel of strength  $q$ . If the defect was instead inducing single-qubit coherent rotations of angle  $\omega$ , one could just twirl them [58], which would indeed generate a dephasing channel of strength  $q = \sin^2(\omega/2)$ .

With this assumption and normal circuit-level noise (Appendix G), we obtain the plot of Fig. 23. From it, one can deduce a threshold value  $q_{\text{th}} = 0.95\%$ , corresponding to a threshold angle  $\omega_{\text{th}} = 0.62$ .

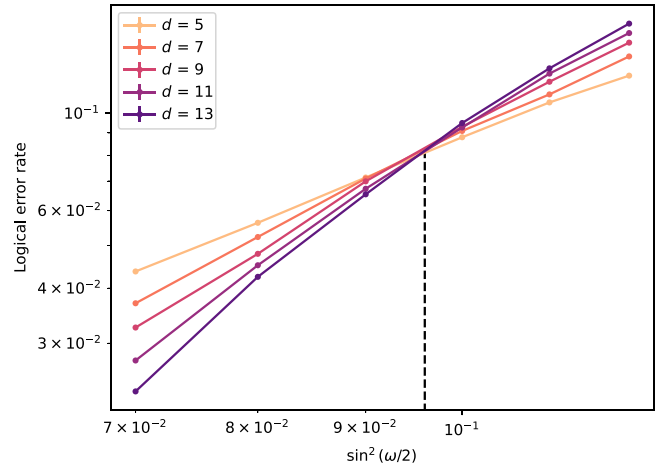


FIG. 23. Defect-induced error threshold. Ancilla and data qubits are subject to circuit-level error of strength  $p = 0.1\%$ . Data qubits additionally experience one round of high errors of strength  $\sin^2(\omega/2)$ , owing to a shuttling event in the vicinity of a defect. The black dashed line indicates the value of the threshold for such error mechanism.

## APPENDIX I: MONITOR QUBITS

### 1. Parameter estimation task

A dominant source of error in spin qubits is the fluctuation of the internal Rabi frequency of the qubits by a variable amount  $\omega$  via the Hamiltonian as  $\mathcal{H} = \omega Z/2$ , where  $Z$  is the Pauli  $Z$  operator. In such a scenario the channel  $\Phi(\cdot) = U(\cdot)U^\dagger$  is unitary and acts on each qubit as a  $Z$  rotation gate by an unknown angle as  $U = e^{-i\omega Z/2}$ .

#### a. Fundamental precision limits

The fundamental limits can be understood in terms of the Cramér-Rao bound which is a lower bound on the variance of the estimator  $\hat{\omega}$  as

$$(\Delta\omega)^2 := \text{Var}[\hat{\omega}] \geq (vF_c)^{-1}, \quad (\text{I1})$$

where  $v$  is the number of times the shuttling event is repeated. Here the classical Fisher information quantifies the sensitivity of the classical probability distribution  $p(q|\omega)$  of measuring the bitstring  $q$  given a particular rotation angle  $\omega$  as

$$F_c = \sum_q p(q|\omega) \left( \frac{\partial \ln p(q|\omega)}{\partial \omega} \right)^2. \quad (\text{I2})$$

Given a fixed input state  $\rho$ , the classical Fisher information is maximized when the best possible measurement basis is used, by mapping the standard basis measurements using the unitary  $\mathcal{U}$  and in this case the quantum Fisher information is obtained which is therefore a general upper bound as  $F_c \leq F_Q$ , please refer to Ref. [59] for more details.

**b. Optimal setting with unentangled monitor qubits**

We first consider the simple scenario when all  $N$  sensing qubits are initialized in a separable pure state. Given the channel  $\Phi$  is assumed to be a unitary  $Z$  rotation, an optimal initial state that maximizes the quantum Fisher information is the  $|+\rangle^{\otimes N}$  state and the optimal measurement that maximizes the classical Fisher information, and thus the achievable precision, is performed in the eigenbasis of the  $X$  Pauli operator. Due to the additivity of the quantum and classical Fisher information one has

$$F_c[|+\rangle^{\otimes N}] = NF_c[|+\rangle] = N,$$

where the second equation follows from  $F_c[|+\rangle] = 1$  via the probability distribution,

$$p(q=0|\omega) = \cos^2\left(\frac{\omega}{2}\right), \quad (I3)$$

$$p(q=1|\omega) = \sin^2\left(\frac{\omega}{2}\right), \quad (I4)$$

by substituting into Eq. (I2). Therefore, the Cramér-Rao bound for this scenario states that the uncertainty  $\Delta\omega$  in learning the rotation angle  $\omega$  is lower bounded as  $\Delta\omega \geq [vN]^{-1/2}$ .

The Cramer-Rao bound is saturated asymptotically by the maximum likelihood estimator but the actual precision may deviate in case of relatively few samples as presently we assume as few as  $N = 200$  samples. Given the likelihood function,

$$\mathcal{L}(\omega) = \cos\left(\frac{\omega}{2}\right)^{2m_0} \times \sin\left(\frac{\omega}{2}\right)^{2(N-m_0)}, \quad (I5)$$

where  $m_0$  denotes the number of times the 0 outcome was observed, one can show that the maximum likelihood estimator is the inverse cosine function of the estimated  $Z$  coordinate of the Bloch sphere as

$$\hat{\omega} = \arccos\left(\frac{2\hat{m}_0}{N} - 1\right). \quad (I6)$$

Since  $\omega$  is estimated from a finite number of shots, our estimates can only take up  $N$  different discrete values as  $\hat{\omega} \in \{\omega_k\}$  via

$$\omega_k = \arccos\left(\frac{2k}{N} - 1\right), \quad 0 \leq k \leq N. \quad (I7)$$

The discrete probability density of  $\omega_k$  can readily be computed given  $\hat{m}_0$  follows a discrete binomial distribution  $B(N, p)$ , where we denote

$$p := p(q=0|\omega) = \cos^2(\omega/2). \quad (I8)$$

This yields the probability  $\mathbb{P}(\omega_k) = \mathbb{P}_{\text{binom}}(k, \omega)$  as the discrete probability density function of the binomial distribution. Furthermore, we compute explicitly the precision

(root mean squared error)  $\Delta\omega$  via

$$\Delta\omega^2 = \sum_{k=1}^N \mathbb{P}_{\text{binom}}(k, \omega) (\omega_k - \omega)^2. \quad (I9)$$

**2. Effect of dephasing and readout errors**

So far we analyzed the precision of ideal state preparation and measurement and now argue that the present scheme is particularly robust against the dominant noise sources in solid-state devices as dephasing and readout errors.

**a. Effect of dephasing noise on precision**

A dominant noise contribution is expected to be due to fluctuations of  $\omega$  faster than the characteristic shuttling times in which case the average effect can be modeled as a dephasing channel. Given that the unitary  $Z$  rotation and dephasing channels commute, this can be viewed as first dephasing the initial state as  $\rho_{\text{deph}} = \mathcal{D}[|+\rangle\langle +|]^{\otimes N}$ , to which a unitary rotation is then applied via  $\Phi$ . In this case the information gain is guaranteed to be lower given that the quantum Fisher information, whose inverse lower bounds the achievable precision, is maximized only by pure states:  $F_Q[\rho_{\text{deph}}] < F_Q[\rho] = N$  where  $\rho$  is the optimal pure initial state we detailed above. In particular, dephasing with probability  $r$  affects our individual single qubit probe states  $\rho_{\text{sing}} = |+\rangle\langle +|$  as

$$\rho' = (1-r)\rho_{\text{sing}} + rZ\rho_{\text{sing}}Z = (1-2r)\rho_{\text{sing}} + 2r\frac{\mathbb{1}}{2}. \quad (I10)$$

Above in the second equation we identify the effect of dephasing on our particular probe state with an equivalent global depolarizing noise channel, for which we can analytically solve the quantum Fisher information [using Eq. (76) in Ref. [59]] as

$$F_Q[\rho'] = \frac{1-4r}{1-\frac{3r}{2}} + \frac{4r^2}{1-\frac{3r}{2}} \geq 1 - \frac{5}{2}r. \quad (I11)$$

Therefore, the QFI of the  $N$ -qubit separable probe state can be computed due to the additivity of the QFI over separable states as  $F_Q[\rho_{\text{deph}}] = NF_Q[\rho'] \geq N(1 - \frac{5}{2}r)$ . Given a sub-surface-code threshold  $r < 0.7\%$  we see that dephasing merely decreases the QFI by a factor  $(1 - \frac{5}{2}r)$  which is only fraction of a percentage. This leads to an increased Cramer-Rao bound, i.e., the dashed lines in Fig. 12 (right) are shifted up via a multiplication by the factor  $(1 - \frac{5}{2}r)^{-1/2}$  due to dephasing. This robustness is thanks to the robustness of the QFI of separable states against decoherence which was studied in detail in Ref. [60]—in contrast, maximally entangled GHZ states are exponentially deteriorated as a function of  $N$ .

Furthermore, probe states have been designed that are optimal against nonunitary channels  $\Phi$  [60] and, in particular, the information gain can be maximized and improved by a constant factor when using entangled squeezed Dicke states as detailed in Ref. [60]. This, however, requires the monitor qubits be prepared in a nontrivial entangled state and due to hardware noise in the state preparation gates, may not be beneficial and therefore our separable states offer a slightly suboptimal but simple alternative.

### b. Effect of measurement errors on precision

For our optimal protocol we assumed perfect projective measurements of the monitor qubits in the relevant bases. However, real measurements are characterized by general positive operator-valued measurements (POVMs) rather than by projections. We illustrate this via a simple readout-error model where entries in the bitstring  $b$ , as measurement outcomes of the  $N$  probe qubits, undergo random and independent bit flips. More specifically, we assume that the  $i$ th bit is flipped as  $0 \rightarrow 1$  with probability  $\alpha_i^+$  and as  $1 \rightarrow 0$  with probability  $\alpha_i^-$ . Therefore, the projective POVM  $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$  on the  $i$ th qubit is replaced by the POVM,

$$\{(1-\alpha_i^+)|0\rangle\langle 0| + \alpha_i^-|1\rangle\langle 1|, \quad \alpha_i^+|0\rangle\langle 0| + (1-\alpha_i^-)|1\rangle\langle 1|\}.$$

For the specific readout-error model when  $\alpha_i^+ = \alpha_i^- =: \alpha_i$ , i.e., the probability is the same for flipping a 1 as flipping a 0, we can easily see that the POVM is identical to performing ideal projective measurements on a dephased probe state  $\rho_{\text{deph}} = \mathcal{D}[|+\rangle\langle +|]^{\otimes N}$  where effective dephasing happens with probability  $\alpha_i$ . Therefore, exactly the same conclusions apply to readout noise as to dephasing noise: as long as the probabilities  $\alpha_i$  are consistent with the surface code threshold, the Cramer-Rao bounds are only increased by a negligible amount due to readout or dephasing errors.

### c. Noisy probability distributions

From Eq. (I10), one can easily update Eq. (I8) to include the effect of dephasing and measurement noise by setting,

$$p' = (1 - 2r)p + r \quad (\text{I12})$$

$$= (1 - 2r) \cos^2(\omega/2) + r. \quad (\text{I13})$$

All probability distributions (binomial laws) can thereon be computed with this new parameter. In all the following plots, we assume an equal level of measurement and dephasing noise at 0.1%, leading to a total noise strength  $r = 0.2\%$ .

## 3. Task (i): Defect detection

After explaining the generalities of monitor qubits, we can now delve into the implementation of the two tasks

monitor qubits are utilized for. The first one is detecting with maximum probability if a defect occurred, regardless of the value of its angle. More precisely, let us denote by  $\omega_{\text{max}}$  the maximum rotation angle the surface code can tolerate to guarantee sufficient error suppression at the chosen code distance, and  $\hat{\omega}_{\text{max}}$  the value of the estimator above which a catastrophic event is declared. As explained in the main text, we want to minimize the rate of false negatives while keeping the rate of false positives around 5%–10%. Both these parameters are controlled by the choice of  $\hat{\omega}_{\text{max}}$ . Besides, we set  $\omega_{\text{max}} = \omega_{\text{th}}/2 = 0.3$  as per Appendix H.

Additionally, we assume that defects (of any angle) are rare and occur at a rate  $\rho$ , and that the distribution of defect angles  $p(\omega)$  is uniform in  $[-\pi, \pi]$ , i.e.,

$$\forall \omega \in [-\pi, \pi] \quad p(\omega) = 1/2\pi. \quad (\text{I14})$$

### a. Rate of false positives

The rate  $P_+$  of false positives is given by

$$P_+ = \rho \times \mathbb{P}(|\hat{\omega}| > \hat{\omega}_{\text{max}} \mid |\omega| < \omega_{\text{max}}) \\ + (1 - \rho) \times \mathbb{P}(|\hat{\omega}| > \hat{\omega}_{\text{max}} \mid \omega = 0).$$

When  $\rho$  is small, one can typically neglect the first term of the equation and assume  $1 - \rho \sim 1$ , leading to

$$P_+ = \mathbb{P}(|\hat{\omega}| > \hat{\omega}_{\text{max}} \mid \omega = 0) \quad (\text{I15})$$

$$= \sum_{k=0}^{k_{\text{max}}-1} \mathbb{P}_{\text{binom}}(k, 0), \quad (\text{I16})$$

where, from Eq. (I7),  $k_{\text{max}} = \lceil N \cos^2(\hat{\omega}_{\text{max}}/2) \rceil$ . The binomial distributions are then computed using Eq. (I12). The result is plotted in Fig. 24 against the number  $N$  of monitor qubits. As expected,  $\hat{\omega}_{\text{max}}$  controls the rate  $P_+$  of false positives. When  $\hat{\omega}_{\text{max}}$  is close to 0, there is a high chance that the estimator  $\hat{\omega}$  falls outside the interval  $[-\hat{\omega}_{\text{max}}, \hat{\omega}_{\text{max}}]$ , even when  $\omega = 0$ . This translates into a high rejection rate, approaching 1. In our case, we are interested in a rejection rate around 5%–10%, which corresponds to the salmon plot, defined by  $\hat{\omega}_{\text{max}}/\omega_{\text{max}} = 0.25$ . We will thus set  $\hat{\omega}_{\text{max}} = 0.075$  in the following.

### b. Distribution of postselected angles

The next step is determining the distribution  $p_{\text{mon}}(\omega)$  of postselected angles, according to the choice of  $\hat{\omega}_{\text{max}}$ . This quantity is given by

$$p_{\text{mon}}(\omega) = p(\omega \mid |\hat{\omega}| \leq \hat{\omega}_{\text{max}}) \\ = \frac{p(|\hat{\omega}| \leq \hat{\omega}_{\text{max}} \mid \omega)p(\omega)}{p(|\hat{\omega}| \leq \hat{\omega}_{\text{max}})} \\ = \frac{p(|\hat{\omega}| \leq \hat{\omega}_{\text{max}} \mid \omega)}{\int_{|\omega|=0}^{\pi} p(|\hat{\omega}| \leq \hat{\omega}_{\text{max}} \mid \omega)d\omega},$$

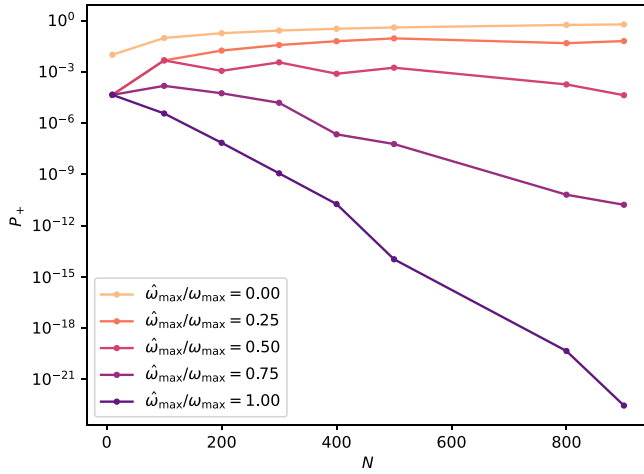


FIG. 24. Probability of a false positive, i.e., declaring that a defect occurred on a nondefective link. See main text for parameters explanation.

where in the last equation we used that the original distribution  $p(\omega)$  of defect rotation angles is uniform. Finally, one gets

$$p_{\text{mon}}(\omega) = \frac{1}{2} \frac{\sum_{k=k_{\text{max}}}^N \mathbb{P}_{\text{binom}}(k, \omega)}{\sum_{k=k_{\text{max}}}^N \int_{\omega=0}^{\pi} \mathbb{P}_{\text{binom}}(k, \omega) d\omega}, \quad (\text{I17})$$

where the integration between 0 and  $\pi$  and the factor 2 arise from symmetry arguments. The result is plotted in Fig. 12 (left). As expected, increasing the number of monitor qubits shrinks the probability distribution around 0. The black dashed lines indicate the value of  $\pm\omega_{\text{max}}$ : when  $\omega$  is outside this window, the chosen code distance does not suffice to adequately lower the logical error rate.

### c. Rate of false negatives

The probability of  $\omega$  falling outside this window is the rate of false negatives, which can lead a logical failure. It is expressed as

$$P_- = \mathbb{P}(|\omega| > \omega_{\text{max}} \mid |\hat{\omega}| \leq \hat{\omega}_{\text{max}}) \quad (\text{I18})$$

$$= 2 \int_{\omega=0}^{\pi} p_{\text{mon}}(\omega) d\omega. \quad (\text{I19})$$

Using  $N = 900$  monitor qubits, meaning that we interleave one monitor qubit per data qubit in a  $30 \times 30$  surface code, one obtains  $P_- = 1.4 \times 10^{-8}$ . This is still computed for a total dephasing and measurement noise strength  $r = 0.2\%$  and  $\hat{\omega}_{\text{max}} = \omega_{\text{max}}/4 = 0.075$ . Therefore, by rejection 5% of the shuttling events, the logical error rate due to defects is brought down to  $1.4 \times 10^{-8} \times \rho$ .

## 4. Task (ii): Estimating large rotation angles

We now focus on task (ii), i.e., the precise evaluation of the angle  $\omega$ . For this purpose, we devise an

estimator that can accurately approximate  $\omega$  in the full range  $(-\pi, \pi)$ , such that we are optimally sensitive to the angles  $0, \pm\pi/2, \pm\pi$ . We therefore initialize the state  $\rho$  of the monitor qubits as  $|\!-\pi/4\rangle^{\otimes N}$ , where  $|\!-\pi/4\rangle := R_z(-\pi/4)|+\rangle$ . Measuring half of them in the eigenbasis of the Pauli  $X$  operator and the other half in the  $Y$  operator results in the probability distribution,

$$p_X(0|\omega) = \cos\left(\frac{\omega - \pi/4}{2}\right)^2,$$

$$p_X(1|\omega) = \sin\left(\frac{\omega - \pi/4}{2}\right)^2,$$

$$p_Y(0|\omega) = \cos\left(\frac{\omega - 3\pi/4}{2}\right)^2,$$

$$p_Y(1|\omega) = \sin\left(\frac{\omega - 3\pi/4}{2}\right)^2.$$

We then use the following estimator:

$$\hat{\omega} = \arctan 2(\hat{x}, \hat{y}) + \pi/4, \quad (\text{I20})$$

which outperforms the maximum likelihood estimator as it makes use of the fact that the true  $x$  and  $y$  coordinates satisfy  $x^2 + y^2 = 1$ . Here  $\hat{x}$  and  $\hat{y}$  are estimators of the  $x$  and  $y$  coordinates of the Bloch sphere, e.g.,  $\hat{x} := (2\hat{x}_0/N/2) - 1$  is the number of times  $\hat{x}_0$  the  $X$  probes resulted in the 0 outcome.

As before the estimator  $\hat{\omega}$  can only take up discrete values as

$$\omega_{kl} = \arctan 2\left(\frac{2k}{N/2} - 1, \frac{2l}{N/2} - 1\right) + \pi/4, \quad (\text{I21})$$

such that  $0 \leq k, l \leq N/2$ . We can then explicitly compute the discrete probability density function given both  $\hat{x}$  and  $\hat{y}$  follow binomial distributions as  $B(N/2, p_x)$  and  $B(N/2, p_y)$  where we denote as  $p_x$  and  $p_y$  the probabilities  $p_X(0|\omega)$  and  $p_Y(0|\omega)$  defined above. Denoting the discrete probability density functions of the two binomial distributions as  $\mathbb{P}_{\text{binom},X}(k)$  and  $\mathbb{P}_{\text{binom},Y}(l)$ , the noiseless probability of the individual  $\omega_{kl}$  estimates is given as

$$\mathbb{P}(\omega_{kl}) = \mathbb{P}_{\text{binom},X}(k) \times \mathbb{P}_{\text{binom},Y}(l). \quad (\text{I22})$$

In the presence of noise, one can easily verify that the arguments of the previous sections still apply. In particular, dephasing and measurement noise are still equivalent, whether the initial state is  $|+\rangle^{\otimes N}$  or  $|\!-\pi/4\rangle^{\otimes N}$ . Besides, the noisy probability distribution can be derived from Eqs. (I12) and (I22), by simply updating the values of  $p_x$  and  $p_y$  as

$$p'_x = (1 - 2r)p_x + q, \quad (\text{I23})$$

$$p'_y = (1 - 2r)p_y + q. \quad (\text{I24})$$

### 5. Test nonunitarity by estimating purity

Unitary rotations can be estimated and then corrected by inverse rotations; however, significant nonunitary noise above threshold can potentially be detrimental. For this reason, we estimate resource requirements for estimating whether the purity of the shuttled qubits remains nearly pure.

One approach for estimating the fidelity is by applying a generalized SWAP test to the qubits in which case we split the  $N$  sensing qubits into batches of  $n$  qubits and for each batch of qubits we apply a SWAP test to estimate  $f = \text{tr}[\rho^n] = 2p_0 - 1$  from the probability  $p_0$  of measuring 0 on the ancilla qubit. The qubits in each batch are initialized identically but can be initialized in different states, potentially random states, across the different batches.

We have overall  $M = N/n$  bits of information to estimate  $f$  via the estimator  $\hat{p}_0 = \hat{m}_0/M$  where  $\hat{m}_0$  is the number of times 0 was observed. In particular, we obtain the estimator  $\hat{f} = 2\hat{m}_0/M - 1$  and the variance

$$\begin{aligned} \text{Var}[\hat{f}] &= \text{Var}[2\hat{m}_0/M - 1] = 4\text{Var}[\hat{m}_0]/M^2 \\ &= 4p(1-p)/M = \frac{1-f^2}{M}, \end{aligned}$$

where we used that  $\hat{m}_0$  is binomially distributed and in the last equation we substituted  $p = 1/2 - f/2$ .

### APPENDIX J: COMPLEMENTARY GAP

One of our defect-detection schemes is based on the computation of the complementary gap corresponding to the measured syndrome. This quantity is given by the length difference between the shortest error strings explaining the syndrome but yielding opposite logical outcomes. Low complementary gaps mean that the decoder's decision cannot be fully trusted.

Similarly to the monitor-qubit case, our aim is to establish a rejection criterion minimizing the rate of false negatives (not detecting a defect) while keeping the rate of false positives around 5% (so as to keep the impact on the compute time moderate). Our selection rule is of the form  $g \geq g_{\min}$ , where  $g$  is the measured complementary gap and  $g_{\min}$  is a distance-dependent parameter. This means that whenever a gap lower than  $g_{\min}$  is measured, the shuttling event is aborted and restarted via snake surgery (Sec. III C).

#### a. Rate of false positives

One can readily adapt the equations of Appendix I and write

$$\begin{aligned} P_+ &= \rho \times \mathbb{P}(g < g_{\min} \mid |\omega| < \omega_{\max}) \\ &\quad + (1 - \rho) \times \mathbb{P}(g < g_{\min} \mid \omega = 0) \end{aligned}$$

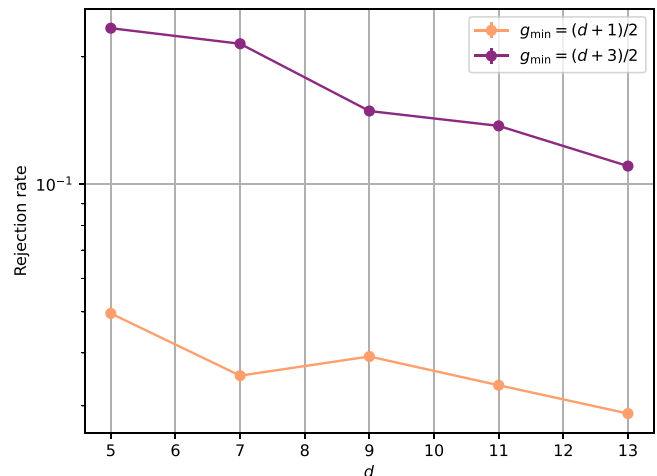


FIG. 25. Rejection rate corresponding to the selection rule  $g \geq g_{\min}$ , where  $g$  is the measured complementary gap and  $g_{\min}$  is a distance-dependent parameter. The simulations are run under circuit-level noise at a rate  $p = 0.1\%$ .

from which it follows

$$P_+ \approx \mathbb{P}(g < g_{\min} \mid \omega = 0), \quad (\text{J1})$$

owing to the assumed low rate of defects (much lower than 5%).

In Fig. 25, we thus plot the rejection rates obtained for two values of  $g_{\min}$  by Monte Carlo simulations, assuming circuit-level noise at  $p = 0.1\%$  and no defect (see simulation details in Appendix G). We observe that setting  $g_{\min} = (d+3)/2$  leads to a rejection rate that is higher than our desired tolerance (around 5%–10%). In contrast, reducing it to  $g_{\min} = (d+1)/2$  yields a rejection rate of 3%–5%, at least for the plotted code distances. This is thus the criterion we will adopt in the rest of the paper. For large code distances, e.g.,  $d = 30$ ,  $g_{\min}$  may be increased to  $(d+3)/2$ . Finally, note that this choice is highly dependent on the physical error rate  $p$ .

#### b. Distribution of postselected angles

The distribution of postselected angles can again be easily obtained by adapting the equations derived for monitor qubits,

$$\begin{aligned} p_{\text{gap}}(\omega) &= p(\omega \mid g \geq g_{\min}) \\ &= \frac{p(g \geq g_{\min} \mid \omega)}{\int_{|\omega|=0}^{\pi} p(g \geq g_{\min} \mid \omega) d\omega}, \end{aligned}$$

$p(g \geq g_{\min} \mid \omega)$  is the acceptance probability given a certain defect angle  $\omega$ . This quantity is obtained by Monte Carlo simulations and the results are plotted in Fig. 14 (left).

**c. Resulting logical error rate**

The main difference between the complementary-gap and the monitor-qubit schemes is that the former additionally shrinks the logical error rate of the postselected instances. This is because rejecting small complementary gaps highly improves the confidence of the decoder thus enhances its performance. We show such error reduction in Fig. 14 (right). The data are obtained with the usual procedure described in Appendix G, the only subtlety being that for every instance of the Monte Carlo simulation, the code is reset and new errors are injected until a complementary gap higher than  $g_{\min}$  is observed.

**APPENDIX K: DEFECT RESISTANCE**

Here, we rigorously estimate the logical error rate *per d stabilizer cycles*  $\tilde{P}_L$  associated with the displacement of a snake from an initial location to a final location in the device. In particular, we aim to express it as a function of  $P_L$ , the static logical error rate per  $d$  stabilizer cycles. This  $P_L$  is the quantity that is normally used as a target error rate to run a predefined algorithm, generally ranging between  $10^{-10}$  and  $10^{-15}$ . However, our architecture extensively makes use of long shuttles, with possible shuttling failures and rerouting due to defects, thereby potentially modifying the raw value of the error rate. In particular, we use a shuttling link deactivation rate of 10%, which is set by hyperparameters in the defect-detection protocols based on monitor qubits and complementary gap.

For this purpose, we assume that, to travel from its initial to its final location, a snake has to attempt a collection of paths, indexed by  $i$ . We then define the following events:

- (a)  $\log$ : a logical error occurred
- (b)  $R_i$ : the  $i$ th path is rejected due to the detection of a defect
- (c)  $\bar{R}_i$ : the  $i$ th path is accepted

The probability of logical error occurring on any of these paths is

$$\begin{aligned} \tilde{P}_L &= \sum_{i \geq 1} \mathbb{P} \left( \log \mid \bigcap_{j=1}^{i-1} R_j \cap \bar{R}_i \right) \times \mathbb{P} \left( \bigcap_{j=1}^{i-1} R_j \cap \bar{R}_i \right) \\ &= \sum_{i \geq 1} ((i-1)\mathbb{P}(\log \mid R_1) + \mathbb{P}(\log \mid \bar{R}_1)) \\ &\quad \times 0.1^{i-1} \times 0.9. \end{aligned}$$

In the first equation, we are decomposing the probability of a logical failure depending on the number of attempted paths (trying  $i$  paths amounts to rejecting  $i-1$  paths and accepting the  $i$ th one). Then, when  $i$  paths are tried, a logical error can occur at any of the  $i-1$  rejections, which is  $\mathbb{P}(\log \mid R_1)$ ; or it can occur on the accepted path, which is

$\mathbb{P}(\log \mid \bar{R}_1)$ . Note that we are here assuming that, for low enough error rates, the probability of two logical errors is negligible. The final factors correspond to the probability of having  $i-1$  rejected paths and 1 accepted path.

Using geometric series formulae, one obtains

$$\tilde{P}_L = \frac{1}{9} \mathbb{P}(\log \mid R_1) + \mathbb{P}(\log \mid \bar{R}_1). \tag{K1}$$

Now,  $\mathbb{P}(\log \mid R_1)$  corresponds to the probability of a logical error given that a defect was detected on the chosen shuttling path. This is handled by the snake-surgery protocol of Sec. III C, where it was shown that the subsequent error rate was that of a normal lattice surgery, i.e., of  $d$  rounds of stabilizer measurements on a static snake. In other words

$$\mathbb{P}(\log \mid R_1) = P_L. \tag{K2}$$

Notably, the accumulated errors stemming from the attempt of multiple paths only represent a small fraction of the normal logical error rate  $P_L$ . This is due to the relatively low rate of false positives we set.

$\mathbb{P}(\log \mid \bar{R}_1)$  however is slightly more complicated to evaluate. It quantifies the probability of a logical failure given that the path was accepted. The main source of error in this case arises from the no-detection of a defect on the accepted path. This probability is thus conditional on the defect rate  $\rho$  (per  $d$  stabilizer cycles) and the performance of the detection strategies. Using the notations from the previous sections, one can write

$$\mathbb{P}(\log \mid \bar{R}_1) = \int_{|\omega|=0}^{\pi} p(\log \cap \omega \cap \bar{M} \cap \bar{G}) d\omega,$$

where  $M$  and  $G$  are events respectively indicating that the monitor qubits or complementary gap rejected a given path.  $\bar{R}_1 = \bar{M} \cap \bar{G}$  as a path is only accepted if it passed both the monitoring and complementary-gap filters. Additionally, we decomposed the probability depending on the defect angle  $\omega$ . Then,

$$\begin{aligned} \mathbb{P}(\log \mid \bar{R}_1) &= \int_{|\omega|=0}^{\pi} \mathbb{P}(\log \mid \omega \cap \bar{M} \cap \bar{G}) \times p(\omega \mid \bar{M} \cap \bar{G}) \\ &\quad \times \mathbb{P}(\bar{M} \cap \bar{G}) d\omega. \end{aligned}$$

The first factor is the probability of a logical failure on a shuttling link that passed all detection tests and that induces a rotation by an angle  $\omega$ . The second factor is the distribution of postselected angles  $\omega$  after both filtering protocols. Finally, the third term is the probability of accepting a path, i.e., 0.9.

The second term contains two contributions, conditional on the presence of a defect or not,

$$p(\omega | \bar{M} \cap \bar{G}) = (1 - \rho)\delta(\omega) + \rho p_{\text{both}}(\omega).$$

Here,  $\delta$  is the Dirac function, indicating that the absence of a defect corresponds to a phase shift  $\omega = 0$ .  $p_{\text{both}}$  is the probability density of postselected angles by both detection techniques in the presence of a defect.

Eventually, one gets

$$\mathbb{P}(\log \cap \bar{R}_1) = 0.9 \times (1 - \rho) \times \mathbb{P}(\log | \omega = 0 \cap \bar{M} \cap \bar{G}) \quad (\text{K3})$$

$$+ 0.9 \times \rho \times \int_{|\omega|=0}^{\pi} \mathbb{P}(\log | \omega \cap \bar{M} \cap \bar{G}) p_{\text{both}}(\omega) d\omega. \quad (\text{K4})$$

The probability in the first term [Eq. (K3)] is the probability of a logical error given that a snake travelled through a nondefective shuttling link that was correctly declared as nondefective. This probability is significantly smaller than  $P_L$ , as the rejection stemming from the complementary-gap scheme highly improves the resulting error rates [Fig. 14 (right)],

$$\mathbb{P}(\log | \omega = 0 \cap \bar{M} \cap \bar{G}) \ll P_L. \quad (\text{K5})$$

The probability appearing in the second term [Eq. (K4)] however corresponds to the error-prone case of an undetected defect and is pondered by the distribution  $p_{\text{both}}$  of postselected defect angles  $\omega$ . Defining

$$P(\omega) = \mathbb{P}(\log | \omega \cap \bar{M} \cap \bar{G}),$$

one can finally write the final form of  $\tilde{P}_L$ ,

$$\tilde{P}_L = \frac{1}{9} P_L + \rho \times \int_{|\omega|=0}^{\pi} P(\omega) p_{\text{both}}(\omega) d\omega. \quad (\text{K6})$$

The first term is the probability that a logical error occurs while rejecting shuttling links via snake surgery, and the second term describes the impact of an undetected defect.

- 
- [1] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Phys. Rev. A* **86**, 032324 (2012).  
 [2] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 h using 20 million noisy qubits, *Quantum* **5**, 433 (2021).  
 [3] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary, [arXiv:quant-ph/9811052](https://arxiv.org/abs/quant-ph/9811052).  
 [4] A. Y. Kitaev, Quantum computations: Algorithms and error correction, *Russ. Math. Surv.* **52**, 1191 (1997).

- [5] D. Bluvstein *et al.*, Logical quantum processor based on reconfigurable atom arrays, *Nature* **626**, 58 (2024).  
 [6] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, Demonstration of the trapped-ion quantum CCD computer architecture, *Nature* **592**, 209 (2021).  
 [7] J. Yoneda, W. Huang, M. Feng, C. H. Yang, K. W. Chan, T. Tantt, W. Gilbert, R. C. C. Leon, F. E. Hudson, K. M. Itoh, A. Morello, S. D. Bartlett, A. Laucht, A. Saraiva, and A. S. Dzurak, Coherent spin qubit transport in silicon, *Nat. Commun.* **12**, 4114 (2021).  
 [8] I. Seidler, T. Struck, R. Xue, N. Focke, S. Trellenkamp, H. Bluhm, and L. R. Schreiber, Conveyor-mode single-electron shuttling in Si/SiGe for a scalable quantum computing architecture, *npj Quantum Inf.* **8**, 100 (2022).  
 [9] M. D. Smet, Y. Matsumoto, A.-M. J. Zwerver, L. Trypuzen, S. L. de Snoo, S. V. Amitonov, A. Sammak, N. Samkharadze, Önder Gül, R. N. M. Wasserman, M. Rimbach-Russ, G. Scappucci, and L. M. K. Vandersypen, High-fidelity single-spin shuttling in silicon, *Nat. Nanotechnol.* **20**, 866 (2025).  
 [10] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, Quantum logic with spin qubits crossing the surface code threshold, *Nature* **601**, 343 (2022).  
 [11] A. Noiri, K. Takeda, T. Nakajima, T. Kobayashi, A. Sammak, G. Scappucci, and S. Tarucha, Fast universal quantum gate above the fault-tolerance threshold in silicon, *Nature* **601**, 338 (2022).  
 [12] A. R. Mills, C. R. Guinn, M. J. Gullans, A. J. Sigillito, M. M. Feldman, E. Nielsen, and J. R. Petta, Two-qubit silicon quantum processor with operation fidelity exceeding 99%, *Sci. Adv.* **8**, eabn5130 (2022).  
 [13] P. Steinacker *et al.*, A 300 mm foundry silicon spin qubit unit cell exceeding 99% fidelity in all operations, [arXiv:2410.15590](https://arxiv.org/abs/2410.15590).  
 [14] J. Yoneda, K. Takeda, T. Otsuka, T. Nakajima, M. R. Delbecq, G. Allison, T. Honda, T. Kodera, S. Oda, Y. Hoshi, N. Usami, K. M. Itoh, and S. Tarucha, A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%, *Nat. Nanotechnol.* **13**, 102 (2018).  
 [15] T. Struck, A. Hollmann, F. Schauer, O. Fedorets, A. Schmidbauer, K. Sawano, H. Riemann, N. V. Abrosimov, L. Cywinski, D. Bougeard, and L. R. Schreiber, Low-frequency spin qubit energy splitting noise in highly purified  $^{28}\text{Si}/\text{SiGe}$ , *npj Quantum Inf.* **6**, 40 (2020).  
 [16] G. Burkard, T. D. Ladd, J. M. Nichol, A. Pan, and J. R. Petta, Semiconductor spin qubits, *Rev. Mod. Phys.* **95**, 025003 (2023).  
 [17] H. Jnane, A. Siegel, and F. Gonzalez-Zalba, Harnessing electron motion for global spin qubit control, [arXiv:2503.12767](https://arxiv.org/abs/2503.12767).  
 [18] A. Siegel, A. Strikis, and M. Fogarty, Towards early fault tolerance on a  $2 \times N$  array of qubits equipped with shuttling, *PRX Quantum* **5**, 040328 (2024).  
 [19] A. Micciche, A. Chatterjee, A. McGregor, and S. Krastanov, Optimizing compilation of error correction codes for  $2 \times N$  quantum dot arrays and its np-hardness, [arXiv:2501.09061](https://arxiv.org/abs/2501.09061).

- [20] Z. Cai, A. Siegel, and S. Benjamin, Lopped pipelines enabling effective 3D qubit lattices in a strictly 2D device, *PRX Quantum* **4**, 020345 (2023).
- [21] M. Künne, A. Willmes, M. Oberländer, C. Gorjaew, J. D. Teske, H. Bhardwaj, M. Beer, E. Kammerloher, R. Otten, I. Seidler, R. Xue, L. R. Schreiber, and H. Bluhm, The SpinBus architecture for scaling spin qubits with electron shuttling, *Nat. Commun.* **15**, 4977 (2024).
- [22] J. M. Boter, J. P. G. Dehollain, J. P. van Dijk, Y. Xu, T. Hensgens, R. Versluis, Henricus W. L. Naus, J. S. Clarke, M. Veldhorst, F. Sebastiano, and Lieven M. K. Vandersypen, Spiderweb array: A sparse spin-qubit array, *Phys. Rev. Appl.* **18**, 024053 (2022).
- [23] R. Li, L. Petit, D. P. Franke, J. P. Dehollain, J. Helsen, M. Steudtner, N. K. Thomas, Z. R. Yoscovits, K. J. Singh, S. Wehner, L. M. K. Vandersypen, J. S. Clarke, and M. Veldhorst, A crossbar network for silicon quantum dot qubits, *Sci. Adv.* **4**, eaar3960 (2018).
- [24] D. Pataki and A. Pályi, Compiling the surface code to crossbar spin qubit architectures, *Phys. Rev. B* **111**, 115307 (2025).
- [25] O. Crawford, J. R. Cruise, N. Mertig, and M. F. Gonzalez-Zalba, Compilation and scaling strategies for a silicon quantum processor with sparse two-dimensional connectivity, *npj Quantum Inf.* **9**, 1 (2023).
- [26] P. Stano and D. Loss, Review of performance metrics of spin qubits in gated semiconducting nanostructures, *Nat. Rev. Phys.* **4**, 672 (2022).
- [27] M. Keppa, N. Focke, Ł. Cywiński, and J. A. Krzywda, Simulation of  $1/f$  charge noise affecting a quantum dot in a Si/SiGe structure, *Appl. Phys. Lett.* **123**, 034005 (2023).
- [28] A. Elsayed, M. M. K. Shehata, C. Godfrin, S. Kubicek, S. Massar, Y. Canvel, J. Jussot, G. Simion, M. Mongillo, D. Wan, B. Govoreanu, I. P. Radu, R. Li, P. Van Dorpe, and K. De Greve, Low charge noise quantum dots with industrial CMOS manufacturing, *npj Quantum Inf.* **10**, 1 (2024).
- [29] J. Yoneda, J. S. Rojas-Arias, P. Stano, K. Takeda, A. Noiri, T. Nakajima, D. Loss, and S. Tarucha, Noise-correlation spectrum for a pair of spin qubits in silicon, *Nat. Phys.* **19**, 1793 (2023).
- [30] H. Jnane and S. C. Benjamin, Ab initio modelling of quantum dot qubits: Coupling, gate dynamics and robustness versus charge noise, *Phys. Rev. Appl.* **24**, 044042 (2025).
- [31] M. M. Shehata, G. Simion, R. Li, F. A. Mohiyaddin, D. Wan, M. Mongillo, B. Govoreanu, I. Radu, K. De Greve, and P. Van Dorpe, Modeling semiconductor spin qubits and their charge noise environment for quantum gate fidelity estimation, *Phys. Rev. B* **108**, 045305 (2023).
- [32] D. Culcer, X. Hu, and S. Das Sarma, Dephasing of Si spin qubits due to charge noise, *Appl. Phys. Lett.* **95**, 073102 (2009).
- [33] J. Rojas-Arias, A. Noiri, P. Stano, T. Nakajima, J. Yoneda, K. Takeda, T. Kobayashi, A. Sammak, G. Scappucci, D. Loss, and S. Tarucha, Spatial noise correlations beyond nearest neighbors in  $^{28}\text{Si}/\text{Si-Ge}$  spin qubits, *Phys. Rev. Appl.* **20**, 054024 (2023).
- [34] M. Mehmandoost and V. V. Dobrovitski, Decoherence induced by a sparse bath of two-level fluctuators: Peculiar features of  $1/f$  noise in high-quality qubits, *Phys. Rev. Res.* **6**, 033175 (2024).
- [35] D. Loss and D. P. DiVincenzo, Quantum computation with quantum dots, *Phys. Rev. A* **57**, 120 (1998).
- [36] J. Levy, Universal quantum computation with spin-1/2 pairs and Heisenberg exchange, *Phys. Rev. Lett.* **89**, 147902 (2002).
- [37] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley, Universal quantum computation with the exchange interaction, *Nature* **408**, 339 (2000).
- [38] A. S. Mokeev, Y.-N. Zhang, and V. V. Dobrovitski, Modeling of decoherence and fidelity enhancement during transport of entangled qubits, *arXiv:2409.04404*.
- [39] V. Langrock, J. A. Krzywda, N. Focke, I. Seidler, L. R. Schreiber, and Ł. Cywiński, Blueprint of a scalable spin qubit shuttle device for coherent mid-range qubit transfer in disordered Si/SiGe/SiO<sub>2</sub>, *PRX Quantum* **4**, 020305 (2023).
- [40] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, *New J. Phys.* **14**, 123011 (2012).
- [41] P. Gokhale, S. Koretsky, S. Huang, S. Majumder, A. Drucker, K. R. Brown, and F. T. Chong, Quantum fan-out: Circuit optimizations and technology modeling, *arXiv:2007.04246*.
- [42] K. Singh, C. E. Bradley, S. Anand, V. Ramesh, R. White, and H. Bernien, Mid-circuit correction of correlated phase errors using an array of spectator qubits, *Science* **380**, 1265 (2023).
- [43] S. Majumder, L. Andreta de Castro, and K. R. Brown, Real-time calibration with spectator qubits, *npj Quantum Inf.* **6**, 1 (2020).
- [44] C. Gidney, M. Newman, P. Brooks, and C. Jones, Yoked surface codes, *arXiv:2312.04522*.
- [45] G. P. Gehér, C. McLauchlan, E. T. Campbell, A. E. Moylett, and O. Crawford, Error-corrected Hadamard gate simulated at the circuit level, *Quantum* **8**, 1394 (2024).
- [46] M. F. Gonzalez-Zalba, S. de Franceschi, E. Charbon, T. Meunier, M. Vinet, and A. S. Dzurak, Scaling silicon-based quantum computing using CMOS technology, *Nat. Electron.* **4**, 872 (2021).
- [47] C. W. Binder and S. C. Benjamin, Shuttling arrives for silicon quantum computers, *Nat. Nanotechnol.* **20**, 857 (2025).
- [48] R. Maurand, X. Jehl, D. Kotekar-Patil, A. Corna, H. Bohuslavskiy, R. Laviéville, L. Hutin, S. Barraud, M. Vinet, M. Sanquer, and S. De Franceschi, A CMOS silicon spin qubit, *Nat. Commun.* **7**, 13575 (2016).
- [49] A. M. J. Zwerver *et al.*, Qubits made by advanced semiconductor manufacturing, *Nat. Electron.* **5**, 184 (2022).
- [50] X. Xue *et al.*, CMOS-based cryogenic control of silicon quantum circuits, *Nature (London)* **593**, 205 (2021).
- [51] A. Ruffino, T.-Y. Yang, J. Michniewicz, Y. Peng, E. Charbon, and M. F. Gonzalez-Zalba, A cryo-CMOS chip that integrates silicon quantum dots and multiplexed dispersive readout electronics, *Nat. Electron.* **5**, 53 (2022).
- [52] S. G. J. Philips, M. T. Madzik, S. V. Amitonov, S. L. de Snoo, M. Russ, N. Kalhor, C. Volk, W. I. L. Lawrie, D. Brousse, L. Tryputen, B. P. Wuetz, A. Sammak, M. Veldhorst, G. Scappucci, and L. M. K. Vandersypen, Universal

- control of a six-qubit quantum processor in silicon, [Nature](#) **609**, 919 (2022).
- [53] K. Takeda, A. Noiri, T. Nakajima, T. Kobayashi, and S. Tarucha, Quantum error correction with silicon spin qubits, [Nature](#) **608**, 682 (2022).
- [54] L. Hutin *et al.*, Gate reflectometry for probing charge and spin states in linear Si MOS split-gate arrays, [arXiv:1912.10884](#) [cond-mat.mes-hall].
- [55] B. M. Maune, M. G. Borselli, B. Huang, T. D. Ladd, P. W. Deelman, K. S. Holabird, A. A. Kiselev, I. Alvarado-Rodriguez, R. S. Ross, A. E. Schmitz, M. Sokolich, C. A. Watson, M. F. Gyure, and A. T. Hunter, Coherent singlet-triplet oscillations in a silicon-based double quantum dot, [Nature](#) **481**, 344 (2012).
- [56] G. Zheng, N. Samkharadze, M. L. Noordam, N. Kalhor, D. Brousse, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, Rapid gate-based spin read-out in silicon using an on-chip resonator, [Nat. Nanotechnol.](#) **14**, 742 (2019).
- [57] K. Takeda, A. Noiri, T. Nakajima, L. C. Camenzind, T. Kobayashi, A. Sammak, G. Scappucci, and S. Tarucha, Rapid single-shot parity spin readout in a silicon double quantum dot with fidelity exceeding 99%, [npj Quantum Inf.](#) **10**, 22 (2024).
- [58] J. J. Wallman and J. Emerson, Noise tailoring for scalable quantum computation via randomized compiling, [Phys. Rev. A](#) **94**, 052325 (2016).
- [59] G. Tóth and I. Apellaniz, Quantum metrology from a quantum information science perspective, [J. Phys. A: Math. Theor.](#) **47**, 424006 (2014).
- [60] B. Koczor, S. Endo, T. Jones, Y. Matsuzaki, and S. C. Benjamin, Variational-state quantum metrology, [New J. Phys.](#) **22**, 083038 (2020).